SONDERHEFT 20 ÖS 100,-/sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 72,- DM 14,-

HAPPY-COMPUTER
2. Atari XL/XE SONDERHEFT

HAPPY-COMPUTER

Markt&Technik

DAS GROSSE HEIMCOMPUTER-MAGAZIN

Atari XL für Kenner

Super Listings

- ★ Turbo Basic für Kassette
- ★ 256 Farben gleichzeitig
- ★ Starke Spiele

Professionell programmieren

- ★ Grundlagen Grafik
- ★ Komplett dokumentierter Registersatz

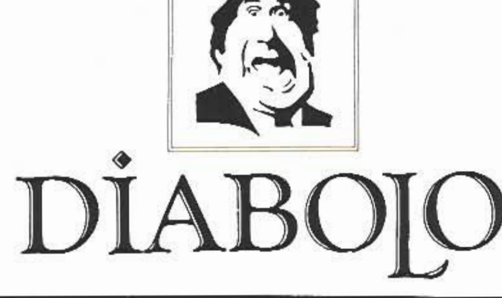
Tolle Tests

- ★ Spiele
- ★ Musik-Compiler
- ★ Hardware-Freezer

ATARI

Alle Programme auch auf Diskette erhältlich







★ Der Versand mit den teuflischen Preisen! ★

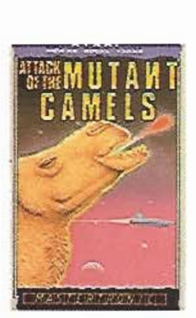
Neue ☎-Nummer: 0 72 52 / 8 66 99



Cass. DM **25.90**
Disk. DM **39.90**



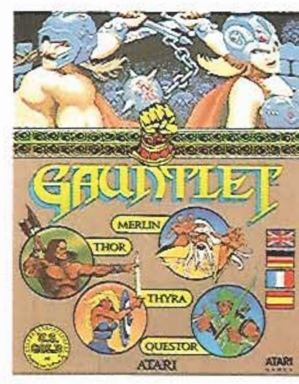
Cass. DM **25.90**
Disk. DM **29.90**



Cass. DM **9.90**



Cass. DM **25.90**
Disk. DM **39.90**



Cass. DM **25.90**
Disk. DM **39.90**

Titel	Cass. / Disk.
Action Biker	9.90 / —.—
Colony	9.90 / —.—
Cristal Rider	9.90 / —.—
Despatch Rider	9.90 / —.—
Gun Law	9.90 / —.—
Hover Bover	9.90 / —.—
Loco	9.90 / —.—
Kik Start	9.90 / —.—
Master Chess	9.90 / —.—
Molecule Man	9.90 / —.—
One Man and his Droid	9.90 / —.—
Vegas Jack Pot	9.90 / —.—
BMX Simulator	14.90 / —.—
Last V.8	14.90 / —.—
LA S.W.A.T.	14.90 / —.—
Ninja	14.90 / —.—
Space Gunner	14.90 / —.—
Spellbound	14.90 / —.—
Red Max	14.90 / —.—
Canon Climber	15.90 / —.—
Designmaster	—.— / 19.80
Shooting Arcade	15.90 / —.—
Actzec	19.90 / 29.90
Boulderdash II	19.90 / 29.90
Cohens Tower/C. Tunnels	19.90 / —.—
Mr. Robot	19.90 / 29.90
Nightrider/Ardy	19.90 / —.—
Spy vs Spy I	19.90 / 29.90
Tale of Beta Lyrae	19.90 / 29.90
Bilbo	—.— / 19.90
Juno First	—.— / 18.90
Max Well's Demon	—.— / 18.90
Mike's Slotmachine	—.— / 19.00
Nibbler	—.— / 18.90
Pyramidos	—.— / 29.00
Tales of Dragons	—.— / 19.00
Soundmachine	—.— / 29.80
Arkanoid	25.90 / 39.90
Space Lobsters	25.90 / 39.90
Sprong	25.90 / 39.90
Starquake	25.90 / —.—
Gauntlet	25.90 / 39.90
Asylum	25.90 / —.—
Boulder Dash Construction Kit	25.90 / 39.90
Fighter Pilot	25.90 / 39.90
Ghostbusters	25.90 / 39.90
Goonies	25.90 / 39.90
Hacker	25.90 / 39.90
Hardball	—.— / 39.90

Titel	Cass. / Disk.
International Karate	25.90 / 39.90
Koronis Rift	—.— / 39.90
Leaderboard	25.90 / 39.90
Montezuma's Revenge	25.90 / 39.90
Polar Piere	25.90 / 39.90
Rescue on Fractalus	25.90 / 39.90
Spy vs Spy II	25.90 / 39.90
Super Huey I	25.90 / 39.90
Tomahawk	25.90 / 39.90
Trailblazer	25.90 / 39.90
Spindizzy	25.90 / 37.90
Grid Runner	9.90 / —.—
Stratosphere	9.90 / —.—
Colossum Chess 4.0	25.90 / 37.90
P.O.D.	9.90 / —.—
Micro Rythm	19.90 / —.—
Power Down	9.90 / —.—
Invasion	25.90 / —.—
Astro Droid	25.90 / 29.90
Footballer of the Year	25.90 / 39.90
Mutant Camels	9.90 / —.—
Frenesis	9.90 / —.—
River Ralley	14.90 / —.—
Bubble Trouble	9.90 / —.—

Doppelpack

Cass. + Disk = **19.90** pro Spiel

Clowns + Ballons, Moonshuttle, Pooyan, Sea Bandits, Spider Quake

S★A★M★P★L★E★R★S

★ Greatest Hits vol. 1

Astro Chase · Bristles
Flip + Flop

DM 29.90/34.90

★ 4 great Games

Jet Set Willy · Balloonacy
Pengon · Wizard

Cass. **DM 19.90**

NEU

- Mercenary Kompendium (dtsch.) 33.90 / 39.90
- The Living Daylights —.— / 39.90
- Head over Heels 25.90 / 39.90
- Auto Duell —.— / 49.–
- Ultima IV —.— / 49.–

Software-Bestellschein

Ich bestelle aus dem Diabolo-Versand folgende Software:

USA No 352–42–CLN

Anzahl	Titel	Gesamtpreis

Computertyp

Name des Bestellers

Anschrift

PLZ/Ort

Datum/Unterschrift

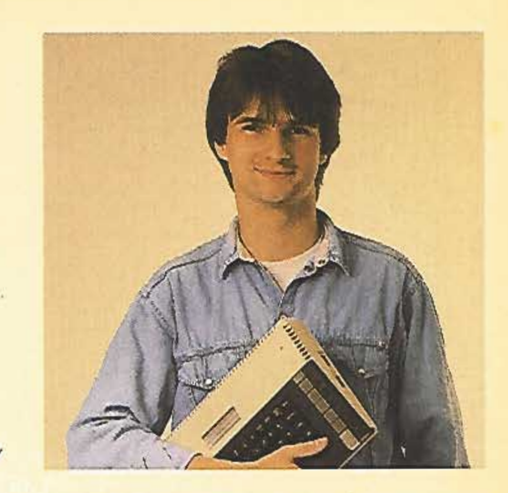
Ich wünsche folgende Bezahlung:

☐ Nachnahme (zuzüglich 5.70 DM Versandkosten)
☐ Vorauskasse (zuzüglich 3.– DM Versandkosten, ab 100 DM Bestellwert versandkostenfrei)

Bei Vorauskasse bitte Scheck beilegen.

Coupon ausschneiden, auf Postkarte kleben und einsenden an:

Diabolo-Versand, PF 1640, 7518 Bretten.
Eine Abteilung des Verlags Rätz-Eberle GdbR.



Henrik Fisch

Totgesagte leben länger

Sie waren sich alle einig, die Softwarehäuser, die Hardwarehersteller und Entwickler, die Händler, die Fachjournalisten und sogar Atari selbst: »Der Atari XL/XE ist tot«. Schließlich war (und ist) der C64 der bislang erfolgreichste Heimcomputer aller Zeiten. Für Atari schien die einzige Zukunft nach den Wirren der Übernahme durch Jack Tramiel in der neuen ST-Linie zu liegen.

Nur noch einige eingefleischte Freaks und ausdauernde User-Clubs hielten ihrem XL/XE die Stange. Die immer kleiner werdende Gemeinde bombadierte unsere Redaktion mit Leserbriefen: »Bringt doch mehr über den Atari XL«. Keine leichte Aufgabe für uns. Wir bemühten uns dennoch, die Wünsche unserer Leserschaft zu erfüllen. Aus der Listingflut der ersten Jahre war allerdings leider ein immer dünneres Rinnsal geworden. Die Produktion des XL/XE wurde eingestellt, die Händler verkauften ihre Restposten nur noch mühsam. Die Preise fielen. Alles schien klar: Der Atari XL ist tot.

Und plötzlich beginnen die Verkaufszahlen zu steigen. Auf der CeBit '87 kann Alwin Stumpf, deutscher Atari-Chef, stolz 92000 verkaufte XL vermelden. Der Erfolg überraschte Atari derart, daß geplant ist, den 800 XL im Gehäuse eines 130 XE neu aufzulegen. Weiterhin werden von Atari Anstrengungen in Richtung Software unternommen. In diesem Jahr wird eine große Werbekampagne für die »kleinen Ataris« erwartet, ein XL-Computer als Spielconsole kommt zu Weihnachten. Der Atari XL lebt!

Im großen Atari-Schwerpunkt in Ausgabe 7/87 von Happy-Computer konnten es die Freaks aus dem berufenen Munde Alwin Stumpfs hören, daß sie an dieser Entwicklung nicht unbeteiligt waren: »Wenn es nicht die Clique der Atarianer gegeben hätte, die ihren Computer leidenschaftlich vor aller Welt verteidigten, wären die kleinen Computer niemals wieder so populär geworden.«

Bereits das erste XL/XE-Sonderheft von Happy-Computer war ein durchschlagender Erfolg. Das »Turbo-Basic« des späteren GFA-Basic-Programmierers Frank Ostrowski (Listing des Monats in Happy-Computer) ist heute Quasi-Standard aller Listings für die 8-Bit-Ataris. Sogar jenseits des Atlantik. Wer regelmäßig amerikanische Atari-Zeitschriften liest, wird dort immer wieder Programme finden, die in Turbo-Basic geschrieben worden sind.

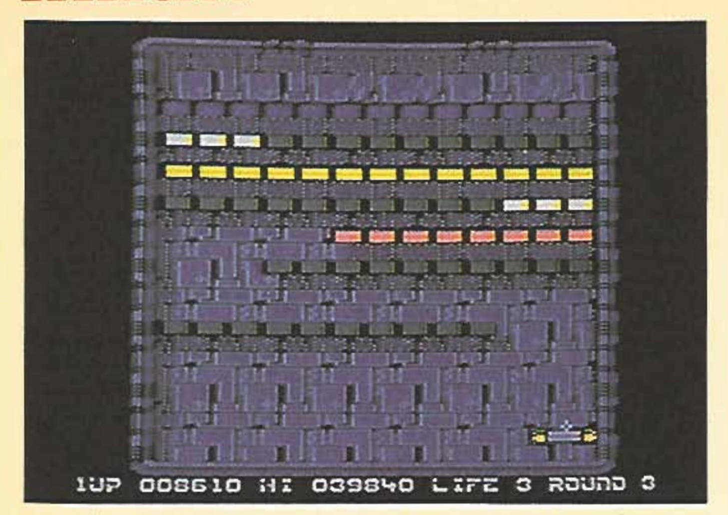
Mit diesem Sonderheft wollen wir dieser Entwicklung noch mehr Durchschlagskraft verleihen. Unter dem Motto »Professionell programmieren« ist dieses Sonderheft ein Nachschlagewerk für alle, die auch noch das Letzte aus ihrem Atari XL herausholen wollen. Wenn Sie die ausführlichste Übersicht über alle XL-Hardware-Register suchen, die je in einer deutschsprachigen Zeitschrift erschienen ist, wenn Sie Grundlagen zur Grafikprogrammierung brauchen, eine Kassettenversion von Turbo-Basic vermissen oder 256 Farben gleichzeitig darstellen wollen: Sie finden es hier.

Für die Anwender unter Ihnen liefern wir unter anderem ein Zeichenprogramm, einige Super-Spiele zum Abtippen und einen Zeichensatzgenerator.

Wenn Sie übrigens ein Spiel oder eine Anwendung programmiert haben, schicken Sie sie uns. Jeden Monat warten in Happy-Computer 3000 Mark auf den Programmierer des »Listings des Monats«.

(Henrik Fisch)

Inhalt



Von jeher wurde der Atari als Spielemaschine eingesetzt. Dieses Talent und seinen Ruf als Spielemaschine hat er sich bis heute bewahrt. Vier neue Spiele für die XL/XE-Computer erwarten Sie ab Seite 6



Nicht nur zum Einfrieren von Programmen, sondern auch als ultimatives Programm-Entwicklungswerkzeug dient der brandneue Turbo-Freezer XL. Einen ausführlichen Test dieses Hardware-Zusatzes finden Sie auf Seite 14



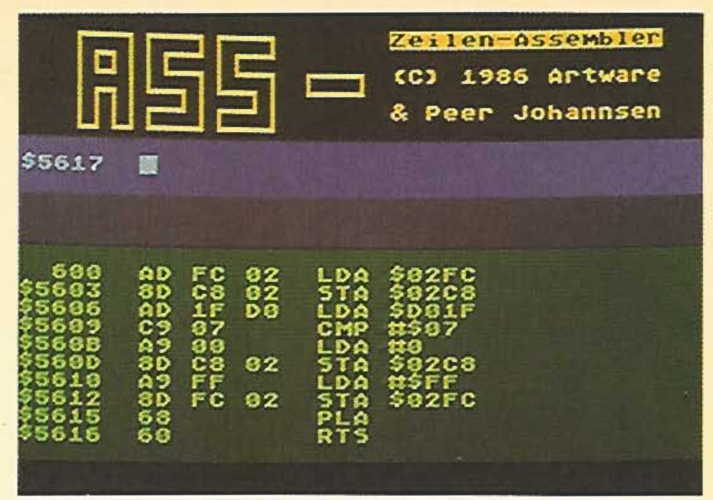
Ein Traum wird wahr: Endlich lassen sich auf dem Bildschirm alle 256 Farben der Atari-Computer gleichzeitig darstellen. Ein Malprogramm und ein Mandelbrot-Programm nutzen die Farbenvielfalt. 21

Spiele

Hardware

Grafik-Listings

Anwendungs-Listings



Mit unserem in Turbo-Basic geschriebenen Zeilen-Assembler lassen sich ohne Probleme kleine Programme schreiben und austesten. Ideal ist er für die Demonstrationsprogramme dieser Ausgabe. 83



Selbstverständlich haben wir wieder Spiele für jeden Geschmack in diesem Heft abgedruckt. Ob Sie nun lieber Action- oder Denkspiele mögen, hier finden Sie hochkarätige Spiele zum Abtippen. 90



Diese Ausgabe ist als handfestes Nachschlagewerk für die Grafik-Programmierung gedacht. Display-Lists, Zeichensätze und verschiedene Interrupts sind keine Rätsel mehr. Zusätzlich erhalten Sie die Registerdokumentation. 120

Spindizzy

Eine neue, gut gelungene Umsetzung macht seit kurzer Zeit von sich reden: Spindizzy. Das Spiel wurde vom Schneider CPC so gut umgesetzt, daß jedem XL/XE-Freak das Herz höher schlagen dürfte.

Weit entfernt, in einer fremden Dimension haben Wissenschaftler eine bizarre Welt entdeckt, die frei im Raum schwebt. Niemand kann sich erklären, woher sie kommt und wozu sie da ist. Fest steht jedoch, daß sie aus lauter geometrischen Formen besteht. Um diese Welt zu erkunden, wird sofort nach einem waghalsigen Mitarbeiter gesucht, der den Flug zu dieser Dimension und eventuell auftretende Gefahren nicht scheut. Da Sie mal wieder knapp bei Kasse sind, melden Sie sich für den Auftrag.

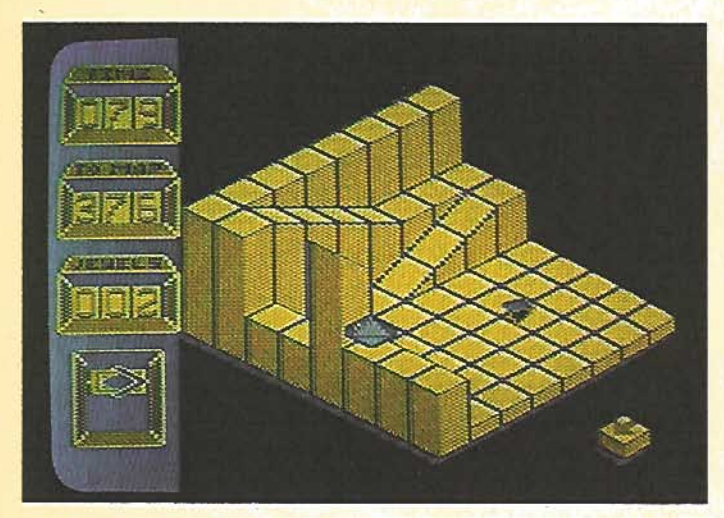
Sie werden entsprechend ausgerüstet und zum Abflughangar gebracht, wo Ihr Raumschiff auf Sie wartet. Als Sie es erblicken, trifft Sie fast der Schlag. Das Raumschiff ist von so alter Bauart, daß Sie glauben, es müsse jeden Moment auseinanderfallen. Es sieht wie eine auf den Kopf gestellte Pyramide aus und besitzt zudem es ein völlig überholtes Antriebssystem. Dieses läßt das Raumschiff ständig um die eigene Achse rotieren, um so den nötigen Schub für eine Vorwärtsbewegung zu erzeugen (wer sich das wohl wieder ausgedacht hat). Widerwillig steigen Sie in das Fahrzeug und fliegen los.

Als Sie bei der Ebene ankommen, zeigt sich Ihnen ein fantastisches Bild. Wie die Wissenschaftler geschildert haben, besteht alles aus geometrischen Formen. Die quadratische Fläche, auf der Sie sich befinden, hat nach allen Seiten Ausgänge und in der Luft schweben schimmernde Diamanten. Als Sie einen der Diamanten berühren, gibt es einen klirrenden Laut, und der Diamant ist verschwunden. Dafür sehen Sie aber, daß sich Ihr Treibstoffvorrat erhöht hat. Insgesamt können Sie auf diese Weise Treibstoff für 150 Sekunden in Ihrem Fahrzeug lagern. Als Sie sich dem Diamanten nähern, stellen Sie eine weitere Unzulänglichkeit Ihres Gefährts fest: Die Bremsen funktionieren nicht. Um zu bremsen, müssen Sie also entweder genau entgegengesetzt zu Ihrer Fahrtrichtung Schub geben oder Sie benutzen die Notbremse (die <SPACE>-Taste am Computer). Mit dieser kommt Ihr Fahrzeug augenblicklich zum Stillstand. Dafür verbraucht diese jedoch bei jeder Betätigung den Treibstoffvorrat einer Sekunde, und Treibstoff ist kostbar. Ohne ihn bleibt Ihr Fahrzeug stehen, und Sie sind mutterseelenallein auf der Ebene gefangen, ohne je wieder zurückzukehren.

Diese Gedanken schieben Sie jedoch entschlossen beiseite und machen sich auf den Weg. Leicht torkelnd und schwankend fahren Sie eine Rinne entlang und finden sich kurz darauf auf einem Gelände wieder, das zum großen Teil mit Wasser überflutet ist. Ein schmaler Weg führt durch die Fluten zum Festland. Gerade rechtzeitig können Sie noch die Notbremse ziehen, um nicht im Wasser zu versinken. Wie nicht anders zu erwarten, kann Ihr Fahrzeug natürlich auch nicht schwimmen. Weiter führt Sie die Reise durch die fremde Landschaft. Bizarr geformte Klippen wechseln mit unebenem Gelände, endlosen Abgründen und Eiswüsten ab. Dann gibt es wieder Gebiete, die an eine Achterbahn erinnern oder die befürchten lassen, in einem Labyrinth ohne Ausgang gefangen zu sein.

Interessant sind auch die verschiedenen Fahrstühle, mit denen Sie in andere Ebenen getragen werden. Einige dieser Fahrstühle lassen sich nur mit in den Boden eingelassenen Schaltern aktivieren, bei manchen sind gleich mehrere dieser Schalter notwendig. Schließlich entdecken Sie auch, daß Sie nicht allein in dieser Welt sind. Unbekannte Wesen huschen in bestimmten Gegenden umher. Schnell merken Sie jedoch, daß diese Wesen es einzig und allein auf Ihren Treibstoffvorrat abgesehen haben. Mit der Zeit fangen Sie an, sich Gedanken über den Sinn dieser Welt zu machen. Wer hat sie geschaffen und welchen Zweck soll sie erfüllen? Wer weiß, vielleicht lüften Sie das Geheimnis.

Spindizzy wurde zuerst für den Schneider CPC geschrieben, dann auf den C64 konvertiert und ist nun auch in einer Version für die Atari-8-Bit-Computer erhältlich. Wenn auch die Grafik keine so detaillierte Darstellung wie auf dem CPC erlaubt, so ist Spindizzy doch eine rundherum gelungene Umsetzung. Einen großen Vorteil hat jedoch die etwa halb so feine Auflösung der XL/XE-Computer: Die Bildschirme werden wesentlich schneller aufgebaut und die einzelnen Objekte bewegen sich fließender. Spindizzy ist seine 29,90 Mark auf Kassette beziehungsweise 39,90 Mark auf Diskette vollkommen wert. (hf)



Wie kommt man an den Diamanten ran?

Steile Klippen machen die Erkundung schwierig

Bücher rund um die Atari ST

O. Hartwig
Atari ST für Insider
1987, 299 Seiten, inkl. Diskette
Systemprogrammierung unter TOS und GEM: Grafikroutinen in C, Quick-Reference-Guide mit BIOS-, XBIOS- und GEM-DOS-Funktionen, Systemadressen und Speicherbelegung, Tips und Tricks.
Best.-Nr. 90423
ISBN 3-89090-423-8
DM 49,–/sFr 45,10/öS 382,20

P. Rosenbeck
C-Programmierung unter TOS/Atari ST
1986, 376 Seiten
Einführung in »C«. Systemprogrammierung am Beispiel eines Diskettenmonitors. Einsatz von BIOS-Routinen. Software-Engineering.
Best.-Nr. 90226
ISBN 3-89090-226-X
DM 52,–/sFr 47,80/öS 405,60



NEU

J. Muus/W. Besenthal
Atari ST Programmierpraxis GFA-BASIC 2.0
1987, 344 Seiten, inkl. Diskette
Tips & Tricks zu 3-D-Grafik, Formular- und Fensterverwaltung, umfangreiches Befehlsverzeichnis, Beschreibung des Compilers, Einbindung von Betriebssystem-Routinen.
Best.-Nr. 90435
ISBN 3-89090-435-1
DM 52,–/sFr 47,80/öS 405,60



P. Wollschläger
Atari-ST-Assembler-Buch
1987, 300 Seiten, inkl. Diskette
Ein 68000-Kurs mit vielen Beispielen. Mit Tips für das Einbinden von Assemblerroutinen in Hochsprachen und ausführlichem Verzeichnis aller GEM-DOS-, BIOS- und XBIOS-Funktionen.
Best.-Nr. 90467
ISBN 3-89090-467-X
DM 59,–/sFr 54,30/öS 460,20



NEU

R. Aumiller/D. Luda
Programmieren mit Forth Atari ST
1987, 531 Seiten, inkl. Diskette
Einführung in Forth. Ausführliche Darstellung der Programmierung unter GEM. Nutzung der Grafikbefehle. Sprites und GEM-TOS-Aufrufe.
Best.-Nr. 90237
ISBN 3-89090-237-5
DM 49,–/sFr 45,10/öS 382,20

P. Wollschlaeger
Atari ST Programmierpraxis ST Pascal
1987, ca. 250 Seiten, inkl. Diskette
Eine strukturierte Anleitung zum professionellen Programmieren unter ST Pascal (Plus). Mit vielen Beispielen für Line-A-Grafik, Sprites, Multitasking, GEM- und maschinennahes Programmieren.
Best.-Nr. 90490
ISBN 3-89090-490-4
DM 59,–/sFr 54,30/öS 460,20

NEU

F. Mathy
Programmierung von Grafik & Sound auf dem Atari ST
1987, 383 Seiten, inkl. Diskette
Vermittelt dem Pascal- und C-Programmierer die Grundlagen zu einer erfolgreichen Grafik- und Soundprogrammierung auf dem Atari ST. Beschreibung der Grafikhardware und eine ausführliche Besprechung der im TOS implementierten Grafikroutinen.
Best.-Nr. 90405
ISBN 3-89090-405-X
DM 52,–/sFr 47,80/öS 405,60

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

Irrtümer und Änderungen vorbehalten.



Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0.
SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656,
ÖSTERREICH: Rudolf Lechner&Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526, Ueberreuter Media Handels- und Verlagsges. mbH Großhandel, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0

COMPUTER-LITERATUR UND SOFTWARE FRÜHJAHR SOMMER 87

Fragen Sie bei Ihrem Buchhändler nach unserem kostenlosen Gesamtverzeichnis mit über 200 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!

Arkanoid

Spielautomaten-Power gibt es auch auf dem Atari! Wer keine müde Mark mehr in die Spielhallen-Automaten investieren will, sollte sich Arkanoid für den XL/XE besorgen.

Die Idee zu dem etwas betagten »Breakout« ist wohl so alt wie die Heimcomputer selbst. Lange Zeit flimmerte das Spielchen in verschiedenen Abwandlungen über die Monitore aller Heimcomputer, bis es nach einiger Zeit in der Versenkung verschwand.

Erst seit kurzer Zeit erlebt die alte Idee ein nie dagewesenes Comeback. Ausgelöst wurde das Breakout-Fieber in der Spielhalle: mit dem Automaten »Arkanoid«. Jetzt ist es endlich auch für die Atari 8-Bit-Computer erschienen.

Die Idee von Arkanoid ist einfach. Sie haben einen Schläger, den Sie am unteren Bildschirmrand hin und her bewegen können. Mit ihm versuchen Sie, einen furios herumsausenden Ball zu treffen. Im oberen Bildschirmdrittel befindet sich eine Mauer, die Sie Stein für Stein abbauen müssen. Jedesmal, wenn der Ball einen Mauerstein erwischt, löst sich dieser (in Wohlgefallen) auf und hinterläßt eine positive Spur auf Ihrem Punktekonto. Wenn Sie alle Mauersteine weggeschossen haben, kommen Sie in einen neuen Level.

Die Hintergrundstory zu Arkanoid ist beachtlich aufgemöbelt, aber eigentlich recht überflüssig für das Spiel: Das Raumschiff Arkanoid wird von einer bösen Macht (von wem sonst...) in einer Raum-Zeitfalle gefangen gehalten. Nur die Raumkapsel »Vaus« hat es geschafft, zu entkommen. Und eben diese Kapsel ist der Schläger, den Sie bedienen. Ihr Ziel ist es, sich durch 32 Screens zu kämpfen, um das Mutterschiff wieder freizubekommen. Damit das richtige Raumschiff-Gefühl aufkommt, ist die Grafik recht abstrakt gezeichnet.

Manchmal sind in den Mauersteinen Zubehörteile verborgen, die man mit dem Schläger einfangen muß. Diese Extras haben die Form von Tonnen, die langsam nach unten fallen. Durch sie kommt Hektik auf. Wenn man nämlich nicht schnell genug ist, kann es passieren, daß man vor lauter Extra-Bonbons den Ball sausen läßt.

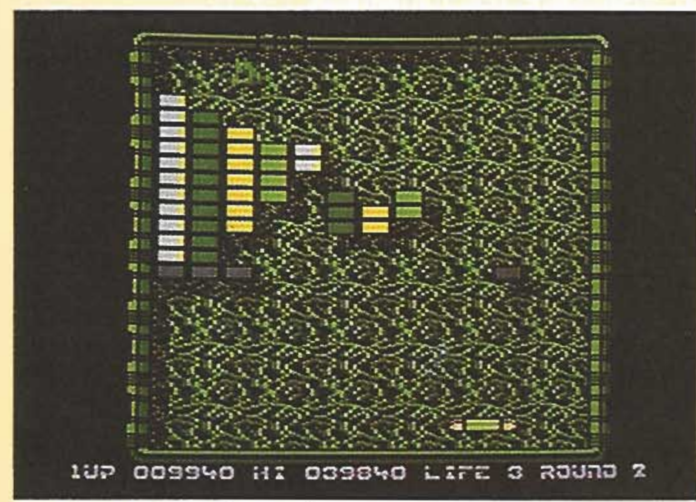
Es gibt insgesamt sieben verschiedene Zusätze, mit denen Sie Ihre Vaus ausstatten können. Ein »P« gewährt ein Extraleben. Wenn Sie ein »G« einfangen, klebt der Ball an dem Schläger und kann so viel präziser wieder abgeschossen werden. Wenn Ihnen zuviel Hektik aufkommt, sollten Sie schauen, daß Sie ein »S« erwischen. Damit wird der Ball langsamer und somit leichter zu treffen. Besonders angenehm ist ein »L«, durch das die Vaus mit einem High-Speed-Laser ausgerüstet wird. Die Mauersteine sind damit in Nullkommanichts abgeräumt.

Das »D« ist etwas für ganz besonders fixe Arkanoid-Spieler. Sobald Sie diese Tonne berühren, verwandelt sich Ihr Ball in drei kleinere Bälle. Natürlich treffen Sie jetzt dreimal so viele Mauersteine, aber Sie müssen auch drei Bälle gleichzeitig im Spiel halten! Mit »E« transformiert sich die Vaus in eine besonders breite Variante. Das wohl begehrteste Extra ist nach wie vor das »B«. Sobald Sie diese Tonne berühren, öffnet sich am rechten unteren Rand die Begrenzung. Wenn Sie mit der Vaus in die entstandene Lücke hineinfahren, kommen Sie automatisch in den nächsten Level. Daß dieses Extra nicht oft in einem Spiel vorkommt, versteht sich wohl von selbst. Oft anzutreffen sind dagegen der Kleber »G« und der »S«low-Modus.

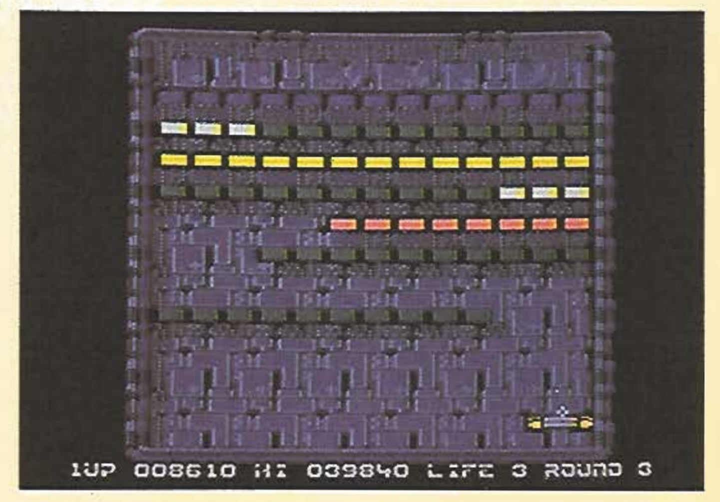
Aber auch mit diesen wichtigen Vergünstigungen werden Sie schwer zu kämpfen haben, denn Arkanoid ist nicht einfach. Spätestens im dritten Level werden Sie Ihre Schwierigkeiten haben. Es tauchen Ziegel auf, die durch nichts zu vernichten sind. Auch ein Laser ist wirkungslos. Sie müssen schon genau zielen, um in den vierten Level zu kommen. Und es stehen Ihnen zweiunddreißig Level bevor ...

Steuern können Sie Ihre Vaus mit so ziemlich allem, was Sie an Ihren Atari anschließen können. Keyboard, Paddles oder Joystick stehen Ihnen zur Verfügung. Die Abfrage ist flott und reagiert gut auf den Spieler.

Grafisch gibt sich Arkanoid auf dem XL/XE etwas zwiespältig. Die Hintergrundgrafik ist absolut dem Automaten nachempfunden und braucht sich nicht vor einem 16-Bitter zu verstecken. Auch die Vaus sieht recht chic auf dem Bildschirm aus. Etwas seltsam ist dagegen der Ball geraten: Vier mal acht Pixel hoch, hat er nicht das beste Aussehen aller Bälle dieser Welt erwischt. Haarsträubend verläuft leider die eckige Flugbahn: Man kann manchmal zusehen, wie der Ball während des Flugs rechnet. Sobald aber die Geschwindigkeit während des Spiels zunimmt, verschwindet der störende Effekt. Insgesamt kann man sagen, daß die Umsetzung für den XL/XE sich nicht hinter anderen verstecken muß. Außerdem ist Arkanoid ein Spiel, von dem man sich nicht mehr so schnell losreißen kann. (al)



Bald haben Sie's geschafft....



...und schon geht's schwierig weiter.



Weitere Programmservice-Angebote für Atari XE/XL

Schnelleres Kopieren auf Atari-Laufwerken

File-Copy: Benötigen Sie ein Programm, mit dem Sie schnell und komfortabel Sicherheitskopien Ihrer Dateien anfertigen können? Wollen Sie Ordnung in Ihre Diskettensammlung bringen? Mit MFCOPY ein Kinderspiel. **Zahlen mit Format:** Mit dieser Routine lassen sich in Turbo-Basic auf einfachste Weise Zahlen runden und formatiert auf den Bildschirm bringen. Diese Funktion ist besonders nützlich für Programme, in denen statistische oder kaufmännische Berechnungen durchgeführt werden. **Tic-Tac-Toe:** Als Endprodukt des Kurses Künstliche Intelligenz selbstgestrickt wurde dieses Programm vorgestellt. Dieses Spiel, bei dem Sie gegen den Computer antreten, veranschaulicht die professionelle Programmierung von Strategiespielen. **Jump:** Sie steuern einen Tennisball, der über eine durchlöcherte Ebene hüpft. Dabei müssen Sie jeden Sprung exakt vorausberechnen und obendrein noch einem Pfeil ausweichen, der sein Unwesen treibt. **The Final Fight:** In diesem fesselnden Action-Spiel müssen Sie Ihr notgelandetes Raumschiff gegen Angriffe verteidigen. Die Bewohner dieses Planeten sehen Ihre Landung als feindliche Kampfhandlung an und versuchen Sie mit allen Mitteln zu vertreiben. Die Anleitung zu diesen Programmen finden Sie in den Ausgaben 4, 5, 6 und 7 der Happy-Computer.
1 Diskette für Atari-XL/XE-Computer

Bestell-Nr.: 20707 **DM 29,90*** sFr 24,90/öS 299,-*

Turbo-Basic – Das Nonplusultra für Programmierer

Turbo-Basic (12/85): Der schnelle Basic-Interpreter für den Atari-Computer. Auf der Diskette befinden sich je eine Version für den Atari 800XL und den Atari 800 mit mindestens 48 KByte RAM. **Ampel (12/85):** Atari-Maschinen-Programm-Eingabe-Listing. **Atari-Prüfsummer:** Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme. **Jumper II:** Listing des Monats 8/84, um die Geschwindigkeit von Turbo-Basic zu demonstrieren. **Magic-Painter:** Listing des Monats 3/85. Ein Zeichenprogramm, das an Turbo-Basic angepaßt wurde.
1 Diskette für Atari 800XL/130XE/800.

Bestell-Nr.: LH8512B **DM 29,90*** sFr 24,90/öS 299,-*

Windows unter Turbo-Basic

Turbo-Windows (9/86): Fenstertechnik für Turbo-Basic. **Drehkörper (9/86):** Ein schnelles Programm in Turbo-Basic zum Zeichnen von 3D-Funktionen. **Labyrinth (3/87):** Unser Listing des Monats März stellt wieder einmal die Krönung der Programmierkunst auf den Atari XE/XL-Computern dar. Sie steuern eine Schlange durch ein Labyrinth und sammeln Gegenstände auf, ohne sich dabei selbst zu beißen. **Morky (1/87):** Schlüpfen Sie in die Rolle unseres kleinen Helden und erforschen Sie insgesamt 35 Räume des Labyrinths, um an Ihr Ziel zu gelangen! **Mord im Computer (4/86):** Als Mark Simpson versuchen Sie, das Verschwinden Ihres Freundes aufzudecken. **Farben gut gemischt (8/86):** Mehr Farben auf dem Bildschirm. **Laufschrift (10/86):** Fehlt Ihnen ein Vorspann für Ihr neues Spiel? Kein Problem mit unserem Maschinenprogramm! **1029 Hardcopy (10/86):** Dieses Programm erledigt schnell und sauber Ausdrucke auf dem Atari 1029-Drucker. **Diashow (11/86):** Zur Präsentation von computergenerierten Bildern. **Nanoconverter (12/86):** Mit diesem Programm können Sie komfortabel und menügesteuert Ihre Spiele von Diskette in den Computer laden. **Drucker-Kosmetik (2/87):** Deutsche Sonderzeichen und Unterlängen sind mit dem Drucker Atari 1029 nicht ohne weiteres darstellbar. Diesem Nachteil wird hier zu Leibe gerückt. **Print-Shop-Grafiken (2/87):** Mit diesem Programm in Turbo-Basic drucken Sie Ihre Grafiken übersichtlich auf Papier.
1 Diskette für Atari-XL/XE-Computer

Bestell-Nr.: 20703 **DM 29,90*** sFr 24,90/öS 299,-*

Grafik unter Turbo-Basic

Turbo-Basic-Interpreter und -Compiler für Atari-Computer • Apfelmännchen in Turbo-Basic • Grafik unter Turbo Basic XL • Daten komprimiert gespeichert • Disksorter • Submission (Labyrinthspiel) • Vorsicht Falle (Action-Spiel) • Pacman mal 2 • Zeichensatzumwandler • Schnellader
2 Disketten für Atari 800XL-/130XE-Computer

Bestell-Nr.: 20703 **DM 29,90*** sFr 24,90/öS 299,-*

Auf der Suche nach dem schwarzen Gold

Ölsuche (8/85): Spiel, bei dem Sie mit dem Atari ST auf Ölsuche gehen. **Atari-Prüfsummer:** Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme. **Geröllheimer (5/85):** Spiel mit Screen-Editor und 20 fertigen Szenen. **24 Farben in Graustufe 0 (6/85):** Routine für farbige Schrift. **Diskhelp (8/85):** Für die schnelle Rettung. **Autostart (9/85):** Basic-Programme starten automatisch. **Dudu 4.0:** Mehr Speicher mit der Floppy 1050.
1 Diskette für Atari 800XL

Bestell-Nr.: LH8510B **DM 29,90*** sFr 24,90/öS 299,-*

Happy-Painter – Das komfortable Malprogramm

Happy-Painter: Unser Listing des Monats der Ausgabe 3/85 ist ein Grafikprogramm, das sich mit anderen Malprogrammen dieser Art messen kann. Besonders gelungen ist die einfache Bedienung, da man mit dem Joystick sowohl im Haupt- als auch in den Untermenüs sämtliche Punkte anwählen kann. Der elektronische Malkasten verfügt über 16 Menüpunkte und bietet eine Grafikauflösung von 160x96 Pixel. **Grafikdemo (3/85):** Alle 256 Farben werden auf dem Bildschirm dargestellt. Eine Farbspielerei, die die hervorragenden Grafikfähigkeiten der Atari-Computer beweist (Rainbow-Effekt). **Variablen-Dump (2/85):** Mit diesem Programm können Sie die verwendeten Variablen eines anderen Programms auf dem Bildschirm listen. **Als die Bilder laufen lernten:** Mit dem Utility Power-Mover können Sie laufende Bilder schnell und problemlos erzeugen. **Statuszeile mit Uhr (1/85):** Damit Sie beim Programmieren nicht die Uhrzeit vergessen, hilft nur eine ständig sichtbare Zeitanzeige. Mit diesem Programm können Sie eine zusätzliche Statuszeile oberhalb des Bildschirms generieren.
1 Diskette für Atari-Computer mit mind. 48 KByte RAM

Bestell-Nr.: LH8503B **DM 29,90*** sFr 24,90/öS 299,-*

Die Jagd nach Gold, Juwelen und Diamanten

Diamantenfieber: Unser Listing des Monats 2/85 ist eine wahre Schatztruhe. Bereichern Sie sich an bunt glitzernden Diamanten, die überall in einem Bergwerk verteilt sind. Aber Vorsicht! Die Stollen sind sehr instabil. Eine falsche Bewegung, und Sie werden von losen Gesteinsbrocken erschlagen. Ein Spiele-Designer sorgt bei unserer exzellenten Boulder-Dash-Variante für anhaltende Spannung. **Die Schatzsuche (Ausgabe 1/85):** Wer möchte sich nicht auch mit einem Schatz bereichern? Wer dazu nicht unbedingt eine Weltreise unternehmen möchte, kann mit seinem Atari 800XL in eine Schatzhöhle eindringen. Gefährliche Tiere wie Skorpione, Ratten und Schlangen erschweren die Suche. **Zeilenzauber (11/84):** Die wichtige RENUMBER-Funktion fehlt leider im Standard-Basic-Editor des Atari. Dieses Programm beseitigt diesen Mangel. **Jumper II (Listing des Monats 8/84):** Ein professionell gemachtes, in Basic geschriebenes Spiel. Exzellente Programmierung, ein High-Score-Zähler und ein eingebautes Demo werden selbst Zweifler schnell überzeugen. Auf musikalische Untermalung wurde großer Wert gelegt, und die verschiedenen Screens sind brillant gemacht. **Mop – Der Goldgräber (7/84):** Schnelligkeit und guter Sound zeichnen dieses Spiel aus. Die Soundfähigkeiten sind wirklich hervorragend. Viele Bilder sorgen für Abwechslung.
1 Diskette für Atari-Computer mit mind. 48 KByte

Bestell-Nr.: LH8502B **DM 29,90*** sFr 24,90/öS 299,-*

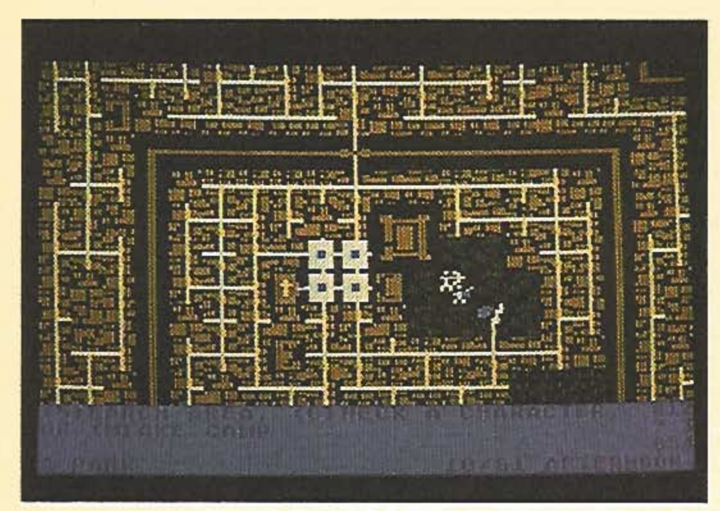
*inkl. Mehrwertsteuer. Unverbindliche Preisempfehlung

Sie suchen packende Spiele, hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.

Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656. Österreich: Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0; Microcomput-ique E. Schiller, Fasangasse 24, A-1030 Wien, Telefon (0222) 785661; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die eingeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Wizard's Crown


Die Kämpfer sind im Stadtpark...



...und müssen sich gegen Riesenspinnen wehren

Ein neues Rollenspiel ist für die kleinen Atari-Computer erschienen: »Wizard's Crown«. Bei diesem Spiel, das ähnlich umfangreich wie »Ultima IV« ist, müssen acht Kämpfer eine Krone suchen.

Es geschah vor 500 Jahren, als der Bund der Zauberer noch über das Land herrschte. Er regierte das Land weise, gerecht und immer zum Vorteil des Volkes. Alle Gerechtigkeit, Güte und Weisheit entstammten einer Krone. Wer diese Krone aufsetzte, besaß alle Weisheit der Welt und konnte Dinge in der Vergangenheit und Zukunft erblicken. Kurz, die Krone war ein absolutes Machtinstrument.

Ein Zauberer war für die Dauer einer vorbestimmten Zeitspanne dazu auserwählt, die Krone zu tragen und das Land zu regieren.

Eines unglücklichen Tages kam es, daß der Zauberer Tarmon an der Reihe war, die Krone an den nächsten im Bunde weiterzuleiten. Der dachte jedoch nicht daran, seine erworbene Machtstellung aufzugeben. So entbrannte ein bitterer Kampf unter den Hexenmeistern, den nur wenige überlebten. Häßliche Kreaturen bevölkern fortan die Ruinen der einst prächtigen Stadt Arghan, von der aus die Magier früher herrschten.

Die letzten Zauberer versuchen verzweifelt, die Krone wieder an sich zu bringen. Kaitar, der älteste der übriggebliebenen, hat dazu einen Plan ersonnen. Er will unerschrockene Abenteurer anheuern, die versuchen sollen, die Krone den Händen Tarmons zu entreißen. Falls sie ihre Aufgabe erfüllen, warten Ruhm und eine so hohe Entlohnung auf die Abenteurer, daß ein Menschenleben nicht ausreicht, das Geld auszugeben.

Soweit zur Vorgeschichte von »Wizard's Crown«, die anderthalb Seiten der 32 Seiten starken Anleitung zum Spiel in Anspruch nimmt. Viel zu lesen also für den Adventure-Begeisterten.

Wizard's Crown ist ein Rollenspiel à la »Dungeons and Dragons«. Soviel wurde in der Vorgeschichte schon verraten: Man spielt mit mehreren Charakteren gleichzeitig. Das ist auch einer der Gründe, warum man sich nicht als Einsteiger an das Spiel heranwagen sollte. Bis zu acht Figuren sollen gleichzeitig kontrolliert werden, wobei die Anleitung nicht ohne triftigen Grund empfiehlt, zu Beginn des Spiels alle acht Charaktere zu nutzen. Das Spiel ist sonst zu schnell vorüber.

Zu Beginn befinden sich alle Charaktere in einer Kneipe, in der sie sich von ihren Streifzügen ausruhen können, erbeutete Schätze und Waffen lagern, sowie im Falle des Ablebens eines Mitglieds einen neuen Abenteurer anheuern können. Von hier aus startet die Kampfgemeinschaft, die Krone zu suchen. Am Anfang sollte man sich jedoch erst einmal auf die Stadt beschränken. Hier gibt es schon genug zu entdecken. Schnell kommt es zu einem ersten Kampf mit einer der zahlreichen Räuberbanden der Stadt. Der Computer stellt dann Kampfmöglichkeiten zur Auswahl: Zum einen gibt es den schnellen Kampf. Hier übernimmt der Computer alle Steuerfunktionen der einzelnen Charaktere und der Angreifer. Zum anderen gibt es den normalen Kampf.

Zu Beginn des Gefechts muß sich die Gruppe erst formieren, um dem Angreifer mit maximaler Kampfkraft gegenüberzutreten. Zauberer sollten sich im Hintergrund halten, und Kämpfer in vorderster Front stehen. Anschließend geht das Hauen und Stechen los. Jeder der acht Gruppenmitglieder kann einen der Angreifer bekämpfen. Dabei spielt die Waffe und Rüstung des Angreifers genau so eine Rolle wie die Position des Gruppenmitglieds zum Feind. Besitzt der Krieger zum Beispiel ein Schild, sollte er sich so hinstellen, daß der Feind links von ihm steht, da er dann einen Schlag des Angreifers mit dem Schild am linken Arm abfangen kann. Insgesamt stehen 21 Funktionen während des Kampfes zur Verfügung.

Haben die Gruppenmitglieder ein Gefecht überstanden, so bekommen sie Erfahrungspunkte, mit denen sie ihre verschiedenen Eigenschaften wie Wendigkeit, Umgang mit der Waffe, Ausdauer und Stärke verbessern. Zauberer brauchen magische Energie, um ihre Kunst betreiben zu können. Nach einem Kampf geht es auch darum, den besiegten Feinden Waffen und Geld abzunehmen. Einige Waffen sind verzaubert und haben eine wesentlich verheerendere Wirkung als die Originalwaffen der Gruppenmitglieder. Waffen, die die Kämpfer nicht gebrauchen können, lassen sich auf dem Markt zu Geld machen. Mit dem Geld wiederum können die Charaktere in einer Kneipe übernachten.

Auch wenn das Spiel extrem kompliziert zu sein scheint (was es übrigens auch ist): Es bringt riesengroßen Spaß. Bis man die Krone gefunden hat und sie Kaitar zurückbringen kann, wird eine ganze Weile vergehen. Und falls man sie wirklich finden sollte, eine Fortsetzung des Adventures ist in Planung. (hf)

Leader Board

Sportspiele gibt es zur Genüge, auch für die kleinen Atari-Computer. Trotzdem hat die Firma Access-Software ihre Golf-Simulation für diesen Computer umgesetzt. Das hat einen guten Grund, denn »Leader Board« ist bislang die am besten gelungene Golfspiel-Simulation, die es zu kaufen gibt.

Bei Leader Board hat man die Wahl zwischen drei Schwierigkeitsstufen, »Novice«, »Amateur« und »Professional«. Bei letzterer spielen Faktoren wie Seitenwind und Anriß des Balls eine tragende Rolle. Diese Stufe sollte man wirklich nur wählen, wenn man schon länger Leader Board gespielt hat und in der Bedienung der vielfältigen Funktionen firm ist. Nach der Wahl des Schwierigkeitsgrades fragt das Programm nach der Anzahl und den Namen der Spieler. Bis zu vier Spieler können gleichzeitig ein Spiel bestreiten, wobei das Programm vier verschiedene Kurse an zu spielenden Löchern zur Auswahl stellt. Kurs 1 ist der leichteste und sollte von dem Anfänger gespielt werden. Bei Kurs 4, dem schwierigsten, sind die einzelnen Bahnen so mit Wasserlöchern und Inseln gespickt, daß es arge Schwierigkeiten bereitet und hoher Genauigkeit bei der Dosierung des Schlages bedarf, um eine Bahn zu meistern.

Das Spiel beginnt damit, daß ein kleines Männchen, zusammen mit der Ansicht der Bahn, aus der Perspektive des Männchens gezeichnet wird. Jetzt legt man mit dem Joystick mittels eines Kreuzes auf dem Bildschirm die Schlagrichtung fest. Ein kleiner Pfahl seitlich des Spielers informiert über die herrschende Windrichtung und die Windstärke. Der Schlag muß nun in dieser Richtung ausgeführt und so dosiert werden, daß der Wind kompensiert wird. Anfangs keine leichte Aufgabe, denn die Kontrollen bei Leader Board sind genauso kompliziert wie funktionsgerecht.

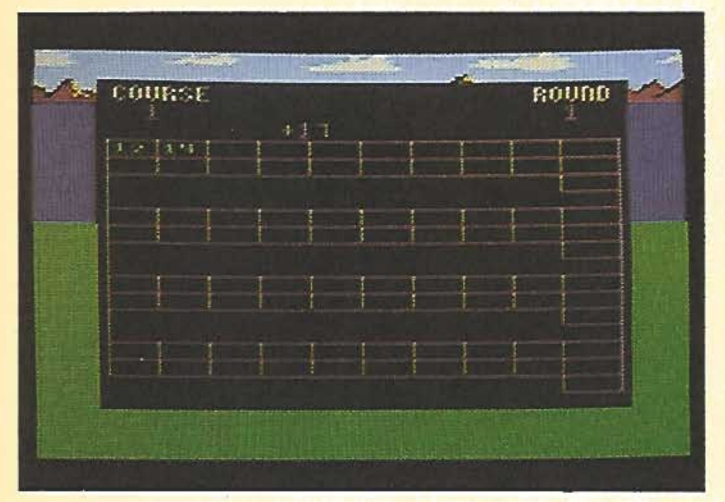
Durch den ersten Druck auf den Joystickknopf legt man die Schlagstärke, durch den zweiten Druck den Anriß des Balls fest. Je nach gewähltem Schläger fliegt der Ball nun eine bestimmte maximale Strecke. Die maximalen Schlagweiten der einzelnen Schläger sind in dem Handbuch aufgeführt. Da der Computer ständig die Entfernung des Spielers zum Loch anzeigt, hat man mit aufgeschlagenem Handbuch neben dem Computer schnell heraus, welchen Schläger man wählen muß. Ging der Schlag nun völlig daneben oder fiel der Ball ins Wasser, so gibt es einen weiteren Versuch von genau der gleichen Position aus.

Ist der Ball dagegen auf dem Festland liegengeblieben, so wird ein neuer Bildschirm gezeichnet. Falls der Ball nun innerhalb eines Radius von 64 Fuß zum Loch gelangt sein sollte, hat man nur noch einen Schläger zur Verfügung. Dieser »Putter« genannte Schläger ist dazu da, den Ball sachte ins Loch zu spielen. Je nach der Anzahl der Schläge, die man für eine Bahn benötigte, werden vom Programm Punkte vergeben. Für jede Bahn ist dabei eine bestimmte Anzahl Schläge vorgegeben. Braucht man weniger Schläge, werden Punkte dazugezählt, bei mehr benötigten Schlägen werden Punkte abgezogen. Es kann also durchaus vorkommen, daß man am Schluß der 18 Bahnen eines Spiels eine negative Punktezahl hat.

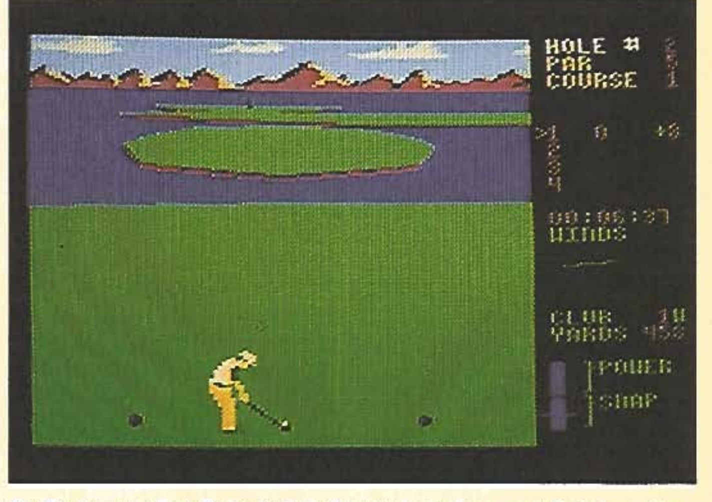
Beim Auspacken der Diskette fällt ein kleiner Stecker auf, der am Joystickport 2 aufgesteckt wird. Dieser Stecker, im Fachjargon »Dongel« genannt, dient als Raubkopierschutz des ansonsten ungeschützten Programms. Das Programm fragt periodisch diesen Stecker ab. Sollte er einmal während des Spiels abgezogen werden, so wird der Bildschirm sofort mit wüsten Linien überzogen, das Programm verweigert in dem Fall die Arbeit. Bei der Diskette befindet sich eine 16seitige Anleitung, in der die besondere Bedienung des Spiels beschrieben wird. Auch wenn den Anfänger die Bedienung des Spiels etwas kompliziert anmutet – hat man das Prinzip der Funktionen erst einmal begriffen, ist die Bedienung kinderleicht.

Wie wohl alle Sportspiele bringt Leader Board mit mehreren Spielern wesentlich mehr Spaß als allein. Da die Simulation durch gute Programmierung einen Bezug zur Realität herstellt, kann man durchaus Geschmack auf richtiges Golfen bekommen. Die Grafik und die Kontrolle über die Schläge ist sehr gut. Leider fehlen wie in anderen Versionen Bäume in den Bahnen. Trotzdem ist Leader Board ein Spiel, das immer wieder gerne hervorgeholt werden wird.

Leader Board ist auf Kassette und Diskette erhältlich und kostet 39 Mark in der Kassetten- sowie 59 Mark in der Diskettenversion. (hf)



Mit der Anzeigetafel hat man immer den Überblick



Golfbahnen aller Schwierigkeitsgrade laden zum Spielen ein

XL tiefgekühlt



Mit einem Freezer lassen sich Sicherheitskopien geschützter Programme anfertigen. Der »Turbo-Freezer XL« bietet zusätzlich ein DOS, einen Monitor mit Disassembler, 256 KByte RAM und das Betriebssystem der alten Atari Computer.

Was ist ein »Freezer«? Ein Freezer (englisch 'to freeze', einfrieren) ist ein Computerzusatz, gleich ob Hardware oder Software, der ein beliebiges Programm im Computer anhält und auf Diskette oder Kassette speichert. Anschließend läßt sich das Programm wieder laden und an genau der gleichen Stelle fortsetzen, an der es »eingefroren« wurde. Auf diese Weise läßt sich von jedem Programm, das nicht während des Laufens Teile nachlädt, eine Sicherheitskopie anfertigen, auch wenn der Kopierschutz noch so gut ausgeklügelt ist.

Für den C64 gibt es Freezer schon länger. Die Atarianer mußten auf diesen nützlichen Zusatz jedoch bislang verzichten, da die interne Struktur der Atari 8-Bit-Computer es sehr kompliziert macht, Programme an einer beliebigen Stelle anzuhalten und ohne Veränderungen wieder zum Laufen zu bringen. Bernhard Engl, der den Atarianern durch sein Turbo-1050-Modul für die Diskettenstation bereits ein Begriff ist, hat nun einen Freezer für den XL entwickelt.

Der Turbo-Freezer XL entpuppt sich als ein 8,5 mal 10,5 Zentimeter großes mit elektronischen Bauteilen vollgepacktes Platinchen, das hinten auf den Erweiterungsbus des XL aufgesteckt wird. An der Seite der Platine steht ein kleines Kabel ab, das die Stromversorgung des Freezers sicherstellt. Ans Ende des Kabels ist eine schmale Hülse angelötet. In der Anleitung wird darauf hingewiesen, daß der Turbo-Freezer XL ganz ohne Lötarbeit zu installieren ist. Deshalb wird die 5-Volt-Versorgungsspannung mit dem Kabel am Joystickport abgegriffen. Die Hülse wird zu diesem Zweck auf den unteren zweiten Pin von links an einem der beiden Joystickports aufgesteckt. Am Computer montiert, sieht der Freezer dann ein wenig abenteuerlich aus, denn die offene Platine steht ein gutes Stück über den Rand des Computergehäuses hinaus, und das Kabel am Joystickport macht auch keinen sehr vertrauenerweckenden Eindruck. Da hätte man sich doch ein stabiles Gehäuse für die Platine gewünscht. Mit Blick auf den Preis des Freezers darf man jedoch nicht meckern.

Wenn man den Computer einschaltet, merkt man zunächst überhaupt nichts vom Freezer. Wir waren im Basic-Modus, und der Computer meldete sich ganz normal mit »READY«. Auf der Freezer-Platine befindet sich ein auffälliger kleiner roter Knopf. Wenn man diesen drückt, erwacht der Freezer zum Leben und übernimmt das Kommando über den Computer. Der Freezer hat jetzt schon sämtliche Prozessor-Register und Hardware-Register gespeichert und die Werte in seinem RAM verwahrt. Man befindet sich dann im Hauptmenü des Freezers. Dieses besteht aus sechs Funktionen, die durch einfaches Tastendrücken aufgerufen werden. Zunächst gibt es die Funktion »Save extern«, mit der der gesamte Computerspeicher auf Diskette oder Kassette gespeichert wird. Bei Diskettenbetrieb läßt sich zusätzlich bestimmen, ob ein vom DOS lesbares File erzeugt werden soll, oder einfach nur die Diskettensektoren beschrieben werden. Um nicht allzuviel Platz zu verschwenden und die Lade- und Speicherzeiten in Grenzen zu halten, wird der Computerspeicher in einer kompakten Form abgelegt. Dieser Datenblock enthält auch sämtliche von Freezer zum Zeitpunkt der Aktivierung gesicherte Registerwerte.

Sicherheit geht vor

Mit »Exec extern« wird ein so gespeicherter Datenblock wieder in den Computer geladen, sämtliche Register restauriert und das Programm an genau der gleichen Stelle fortgesetzt, an der es unterbrochen wurde. Wir haben einige Programme mit dem Freezer ausprobiert und konnten keins finden, das nach dem Auftauen nicht lief, oder das nach dem Starten irgendwelche Fehler zeigte. Mit dem Freezer hat man also die Möglichkeit, von kopiergeschützten Programmen Sicherheitskopien zu ziehen. Interessant ist auch, ein Spiel an einer gefährlichen Stelle einzufrieren und zu speichern. Verliert man sein Spielerleben, kann man mit dem Freezer sofort an der gespeicherten Stelle weitermachen und muß sich nicht erst wieder durch sämtliche Gefahren kämpfen.

Mit dem dritten Menüpunkt, »Zero OS RAM«, wird das RAM unter dem Betriebssystem gelöscht. Programme, die diesen Speicher nicht benutzen, werden dadurch in ge-


Markt&Technik
2/87
68000er
DAS MAGAZIN DER NEUEN COMPUTER-GENERATION
FEBRUAR 1987
DM 7,-
Atari ST
• Wichtige Kaufhilfen für Einsteiger
• Vorgestellt: die besten Programme
• Riesen Hardware-Überblick
• Multitasking mit GfA-Basic
• Macintosh-Emulatoren im Test
Amiga
Grafik-Sensation Deluxe Paint II
QL
Große Assembler-Parade
Mac
kreativ!
Interview mit »Axel F.« – Komponist Harold Faltermeyer
Viele anspruchsvolle Listings für Atari ST, Amiga und QL
AUFBRUCH IN EINE NEUE DIMENSION
mit »68000er«, dem Magazin der neuen Computer-Generation
▶ Programmiersprachekurse für Basic, C, Modula und Assembler.
▶ Bauanleitungen für professionelle Hardware-Erweiterungen.
▶ Spiele-Spaß und -Spannung auf höchstem Niveau.
Ihre hot-line zur Spitzentechnologie von AtariST, Amiga, Macintosh und Sinclair QL.
Das »68000er«-Magazin erscheint
jeden Monat neu!
POSTER & GUTSCHEIN
KOSTENLOS FÜR SIE
84 mal 60 Zentimeter High-Tech-Szene erwarten Sie! Ihr »68000er«-Poster ist im Abonnementpreis enthalten und gehört Ihnen, auch wenn Sie Ihre Bestellung widerrufen sollten.
FÜR EIN KOSTENLOSES PROBEEXEMPLAR DES »68000er«-MAGAZINS
JA, ich möchte »68000er«, das Magazin der neuen Computer-Generation, kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »68000er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte es dann regelmäßig frei Haus per Post. Außerdem nutze ich den Abonnement-Preisvorteil von 8 % und bezahle pro Jahr nur 77,– DM statt 84,– DM im Einzelverkauf.
Vorname
Name
Straße
PLZ, Ort
Datum
1. Unterschrift
Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.
Datum
2. Unterschrift
Gutschein ausfüllen und absenden an: Markt & Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar
HCS20



ATARI

dBMAN

Das relationale Datenbanksystem für Ihren Atari ST

dBMAN ist ein leistungsstarkes und flexibles Datenbanksystem

Ihre Daten werden damit relational aufgebaut und verarbeitet. Doch das ist noch nicht alles: Mit der integrierten, bedienerfreundlichen Programmiersprache von dBMAN entwickeln Sie Ihre individuellen Anwendungsprogramme. Sie erstellen Auftragsbearbeitungsprogramme und andere komplizierte Anwendungen. dBMAN ist zu dBASE III kompatibel.
Als Umsteiger haben Sie dadurch die Möglichkeit, sofort mit dBMAN zu arbeiten.

Leistungsmerkmale:

- O dBMAN erzeugt vollständige Datenbanken. Sie können auch im nachhinein ohne Datenverluste die Struktur verändern und erweitern.
- O Sie verwalten Ihre Daten mit dem mitgelieferten ASSIST(enten), ohne zeitraubend ein Handbuch wälzen zu müssen.
- O Das Zufügen, Löschen, Verändern, Darstellen und Ausdrucken von Daten erfolgt auf einfache Weise.
- O Sie können bis zu 10 Dateien gleichzeitig öffnen, 2 Billionen Datensätze pro Datei verwalten und eine unbegrenzte Zahl von Variablen verarbeiten.



Lieferumfang:

- O umfangreiches deutsches Handbuch
- O 3½"-Diskette
- O Beispieldiskette

Hardware-Anforderung:

Atari ST mit mindestens 400 Kbyte freiem Speicher (1Mbyte RAM oder 0,5 Mbyte und TOS im ROM), SM-124- oder SC-124-Monitor, mindestens 1 Diskettenlaufwerk.

Bestell-Nr. 51109
DM 399,-* (sFr 345,-/öS 3990,-*)

* inkl. MwSt. Unverbindliche Preisempfehlung

Diese Markt&Technik-Softwareprodukte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder im Buchhandel. Wenn Sie direkt beim Verlag bestellen wollen: gegen Vorauskasse durch Verrechnungsscheck oder mit der abgedruckten Zahlkarte.

	Version	Best.-Nr.	Format	Preis DM	sFr	öS
dBMAN (Vers. 3.00G)	Atari ST	51109	3½"	399,-*	345,-	3990,-*
Update zu dBMAN (Vers. 3.00G)	Atari ST	51109U	3½"	98,-*	98,-	980,-*
WordStar/MailMerge	Atari ST	50106	3½"	199,-*	178,-	1990,-*
dBASE II	Atari ST	50306	3½"	348,-*	295,-	3490,-*
Update Protext 1.X auf 2.1	Atari ST	51644	3½"	30,-*	30,-	299,-*
Protext 2.1	Atari ST	51643	3½"	148,-*	132,-	1490,-*

* inkl. MwSt. Unverbindliche Preisempfehlung



Zeitschriften · Bücher
Software · Schulung

707215

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

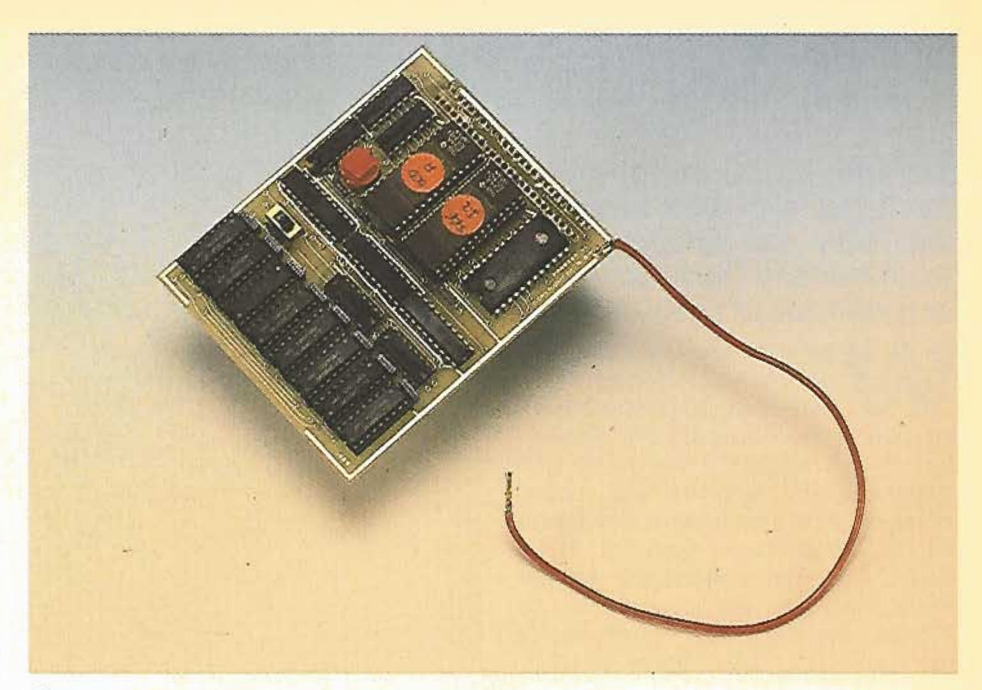
Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

freezter und gespeicherter Form wesentlich kürzer.

Der vierte Menüpunkt speichert ein eingefrorenes Programm nicht auf Diskette oder Kassette, sondern auf einer RAM-Disk. Auf der Freezerplatine sind zu diesem Zweck acht Sockel für RAM-Chips freigelassen, mit denen der Freezer in Schritten von 64 KByte auf 256 KByte aufgerüstet werden kann. Mit 64 KByte RAM auf der Platine hat man also den Speicher eines 130 XE zur Verfügung. Nach der Anleitung ist man damit jedoch nicht ganz 130-XE-kompatibel. Im 130 XE werden die zusätzlichen 64 KByte RAM in vier 16-KByte-Bänken in den Speicher eingeblendet. Der XE läßt sich nun so programmieren, daß der Mikroprozessor und der Bildschirmprozessor auf unterschiedliche Seiten zugreifen können. Falls ein Programm diese Betriebsart ausnutzt, wird es nicht ohne weiteres mit der RAM-Erweiterung des Turbo-Freezers XL zusammenarbeiten. Zum einen gibt es jedoch bisher kein kommerzielles Programm mit dieser Betriebsart. Zum anderen ist in der Anleitung sehr genau beschrieben, wie man ein Kabel im Inneren des Computers so an den Erweiterungsbus anlötet, daß das Umschaltsignal zur Verfügung steht. Doch zurück zum Freezer-Menü. »Save Ramdisk« speichert wie »Save extern« den Computerspeicher mit allen Registerwerten. Das Gegenstück dazu ist der Menüpunkt »Exec Ramdisk«, der aus der RAM-Disk ein Programm lädt, auftaut und startet.

Register eingefroren

Der sechste und letzte Menüpunkt läßt ahnen, welches Potential in dem Turbo-Freezer XL steckt. Wählt man den Menüpunkt »DOS/Debugger« an, hat man einen kompletten Monitor sowie ein komplettes DOS zur Verfügung. Mit dem Monitor lassen sich dann endlich die tiefsten Geheimnisse von Profiprogrammierern lüften. Für Leute, die beim Anblick von Programmen wie »Rescue on Fractalus« in andächtiges Staunen verfallen und bei denen sich im Kopf riesengroße, auf den Kopf gestellte Fragezeichen bilden, wenn sie gefragt werden, wie man so etwas programmiert, ist diese Funktion ideal. Ohne Probleme kann man nun ein in voller Aktion angehaltenes Programm untersuchen. Zusätzlich schaltet der Freezer automatisch das RAM unter dem Betriebssystem ein, damit man sich die dort stehenden Programme anschauen kann. Weiterhin ist der Freezer so konstruiert, daß das eigentliche Freezer-Programm, sozusagen das Betriebssystem des Freezers, nicht sichtbar ist und auch keinen Speicherplatz im Computer belegt. Ein Programm bemerkt also niemals den Freezer und hat somit auch keine Möglichkeit, Gegenmaßnahmen einzuleiten. Dem Freezer und dem Monitor ist jedes Programm schutzlos ausgeliefert. Nach Belieben läßt sich ein Programm disassemblieren, verändern oder entschützen. Für eigene Programme ist der Monitor das ultimative Entwicklungswerkzeug, da sich jedes Programm sofort beim Auftauchen eines Fehlers im Fehlerzustand untersuchen läßt.



Der Freezer Prototyp. Die Serienmodelle werden mit Lötstopmaske geliefert

Das eingebaute DOS stellt einen weiteren Knüller des Freezers dar. Ein Knopfdruck genügt, und es läßt sich mal eben schnell eine Diskette formatieren. Dabei gibt es jedoch einen kleinen negativen Kritikpunkt. Jedes normale DOS schreibt nach dem Formatieren einer Diskette auf die ersten drei Sektoren ein Programm, das auf dem Bildschirm »Boot Error« ausgibt, wenn von der Diskette gebootet wird und sich kein DOS darauf befindet. Der Freezer tut dies nicht, weshalb sich der Computer aufhängt, wenn man versucht, von einer Freezer-formatierten Diskette zu booten. Weitere Folgen hat dies keine, es ist halt nur nicht besonders schön.

Selbstverständlich werden Disketten in Single Density (90 KByte), Medium Density (130 KByte) und Double Density (180 KByte) unterstützt. Letztere allerdings nur mit einem aufgerüsteten Laufwerk. Neben den Befehlen zum Formatieren von Disketten stehen Kommandos zum Anzeigen des Inhaltsverzeichnisses einer Diskette, zum Schützen, Entschützen und Umbennen von Files sowie zum Laden und Speichern von Daten zur Verfügung.

Auf der Freezer-Platine ist neben dem EPROM mit der Betriebssoftware des Freezers noch ein weiterer Sockel frei. Gleichzeitig befindet sich auf der Platine über dem roten Knopf ein Schiebeschalter. Blättert man in der Anleitung, so wird der Sinn des freien Sockels schnell klar: Hier läßt sich ein EPROM mit dem Betriebssystem der alten 400er/800er Computer von Atari einsetzen. Es gibt einige Programme, die mit dem Betriebssystem der neuen Computer nicht funktionieren. Da kommt das alte Betriebssystem im ROM sehr recht, denn damit hat man quasi einen alten Atari Computer und ab sofort keine Probleme mehr mit inkompatibler Software.

Bei aller Euphorie hat der Freezer jedoch einen Nachteil: Er läßt sich nur am Atari 800XL betreiben, die Besitzer eines 130XE müssen vorerst noch auf eine Adapterplatine warten.

Das alte Betriebssystem sowie die RAM-Disk in beliebiger Ausbaustufe kann gleich mitgeordert werden. Dazu schreibt man an:

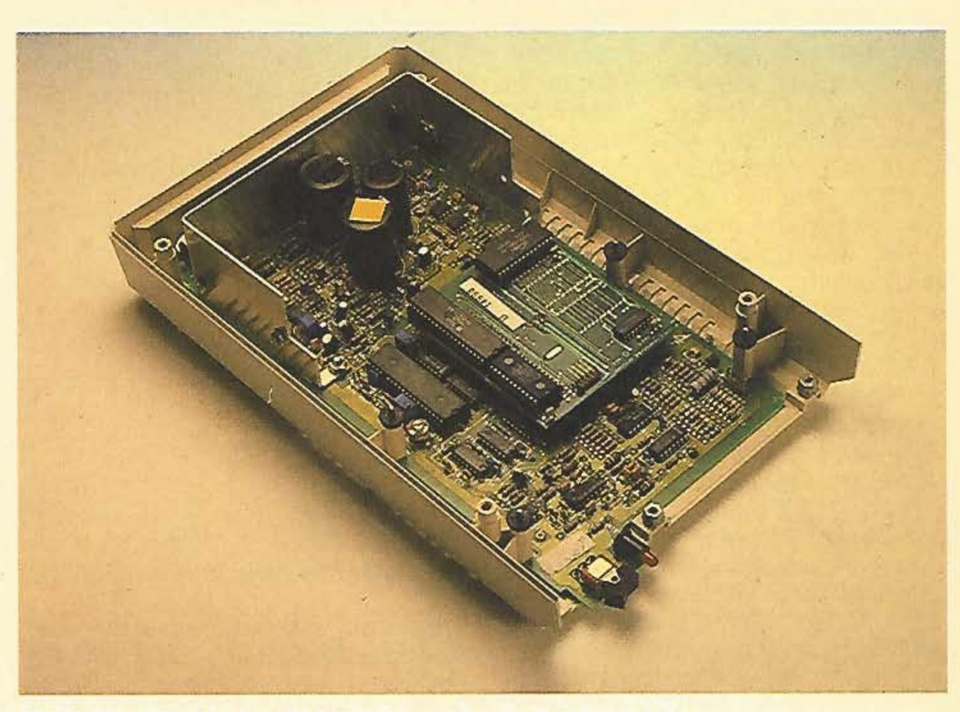
Gerald Engl
Kennwort: Turbo-Freezer XL
Bunsenstraße 13
8000 München 83

Der Freezer kostet 149 Mark, das alte Betriebssystem zusätzlich 10 Mark und jede 64-KByte-Stufe der RAM-Disk 20 Mark. Wenn man sich anschaut, was der Freezer alles leistet, und dann seinen Preis mit dem von Konkurrenzprodukten vergleicht, kann man nur sagen, daß der Freezer ein echter Hammer ist. (hf)

Ein Laufwerk wird fröhlich

Schneller, besser, komfortabler. Damit läßt sich kurz beschreiben, was das »Happy Drive Enhancement« aus der 1050-Diskettenstation herausholt.

Was wäre ein Computer wie der Atari XL ohne eine Diskettenstation? Doch Disketten-Operationen bezahlt der Anwender oft mit langen Wartezeiten und umständlichem Kopierschutz. Diese Tatsache ließ die amerikanische Firma Happy Computers schon im Jahre 1982 auf den Gedanken kommen, ein Hardware-Utility auf den Markt zu bringen, mit der diese Nachteile für den Benutzer aufgehoben werden. Die Entwickler nannten ihre Erweiterung »Happy Drive Enhancement«. Diese Idee war von so durchschlagendem Erfolg, daß sie in Deutschland einige Nachahmer wie »Speedy 1050« oder »Turbo-Modul« nach sich zog. Doch an das Original reichten diese Nachfolger nicht heran. Ein Grund für uns, den »Ur-Speeder« etwas genauer anzusehen.



Eine kleine Platine im Laufwerk macht's möglich

Kurze Ladezeiten

Das Happy Drive Enhancement, das den Platz des ROM und den des Prozessors in der Diskettenstation einnimmt, versetzt seinen Besitzer in die Lage, den Controller direkt zu programmieren und damit die Diskettenstation vielseitiger zu handhaben, als es normalerweise möglich ist.

Auf der aufzusteckenden Platine sind neben ein paar Logikbausteinen ein 8-KByte-EPROM, ein 8-KByte-RAM sowie eine 6502 CPU untergebracht – ein vollständiger kleiner Computer also. In dem RAM werden nicht nur die Übergabeparameter des Computers gespeichert, sondern es dient auch dazu, immer einen Track vollständig einzulesen. Dadurch wird eine geringere Belastung der Mechanik und vor allem eine erhebliche Geschwindigkeitssteigerung erreicht, da nachfolgende Sektoren schon zur Verfügung stehen und nicht mehr extra gelesen werden müssen.

Ein vom Anwender benutztes normales DOS merkt nichts von diesen Vorgängen, da sie alle von dem Happy Drive Enhancement gesteuert werden. Weiterhin ist es möglich, eigene Programme im Happy Drive Enhancement laufen zu lassen und so die Diskettenstation fest im Griff zu haben.

Der Benutzer kann seine Disketten mit dem Enhancement für die Atari 1050 auf Double-Density formatieren, was ihm dann 176 KByte auf jeder Diskettenseite zur Verfügung stellt. Den Besitzern eines alten Laufwerks, der Atari 810, steht die doppelte Schreibdichte nicht zur Verfügung, da der dort eingebaute Controller dieses Format nicht erzeugen und verwalten kann. Wegen der unterschiedlichen Controller-Typen (bei der Atari 810 ist es der 1771, bei der Atari 1050 der 2793, beide von Western Digital) gibt es zwei verschiedene Happy Drive Enhancements, die Software ist jedoch identisch.

Ein weiterer Vorteil des Speeders: Der Benutzer ist nicht mehr auf die Übertragungsrate von 19200 Baud (übertragene Bits pro Sekunde) zwischen Computer und Diskettenstation angewiesen, sondern kann mit bestimmten Software-Einstellungen bis zu 70000 Baud erreichen. Damit wird aus dem monotonen Piep-Piep beim Laden und Speichern ein hohes Pfeifen.

Verzichtet man hierauf, kann man selbst in der normalen Betriebsart die Geschwindigkeit steigern, indem man ein DOS erzeugt, bei dem Verify-Write ausgeschaltet ist (mit POKE 1913,80). Das Happy Drive Enhancement führt diese Kontrolle schon selbständig durch.

Zusätzlich zu der Platine werden das »Happy Warp Speed Menü« und das »Warp Speed Dos« mitgeliefert, das mit dem Atari Dos 2.0 zusammenarbeitet.

Das Warp-Speed-Menü erscheint mittlerweile in der Version 7.1, die dem Benutzer neun Menüpunkte zur Verfügung stellt:

Drive Options:

Hier kann der Anwender

- sein Laufwerk wieder zu einem gewöhnlichen Atari-Diskettenlaufwerk ohne Beschleuniger machen, damit manche heiklen Original-Programme laufen. Sämtliche Funktionen des Happy Drive Enhancement sind dann abgeschaltet. Nur nach einem Aus- und wieder Anschalten wird das Atari-Laufwerk wieder »fröhlich«.
- das Happy Drive Enhancement initialisieren, um vorher eingestelle Optionen wieder aufzuheben.
- den »Drive-Select« außer Kraft setzen und Laufwerk 1 beispielsweise zu Laufwerk 2 machen.
- die Zeit bestimmen, in der das Laufwerk nach dem letzten Zugriff noch laufen soll, um bei einem eventuell erneuten Zugriff die Anlaufzeit des Motors zu sparen.
- auf schnelles Schreiben schalten.
- den U.S. Emulator installieren. Kopierprogramme beispielsweise, die diesen ansprechen, schreiben und lesen dann um ein Vielfaches schneller als ohne den Emulator.
- Laufwerk auf Write-Enable stellen.
- Laufwerk auf Write-Protect stellen.

Diagnostic:

Hier wird das »Happy Enhancement Diagnostic«-Programm nachgeladen.

Es folgt eine Auswahl von Test-Funktionen: RAM und ROM der Erweiterungen werden genauso überprüft wie Übertragungsrate, Anzahl der Umdrehungen pro Minute, Kopfpositionierung, Schreib- und Lesesicherheit und die zusätzliche Hardware-Option.

Sector Copier:

Ein Diskkopierer wird nachgeladen, mit dem Atari-Disketten auf sehr komfortable Art und Weise dupliziert werden können. So erlaubt es die Erweiterung hier, Start- und Endsektor des zu kopierenden Diskettenbereichs einzugeben.

Happy Backup:

Der Diskettenkopierer ist das Herzstück dieses Menüs. Dieses nachzuladende Programm kann allein oder mit Hilfe sogenannter »PDP-Files« fast alle kopiergeschützten Single-Density-Disketten duplizieren. PDP-kopierte Disketten laufen dann nur mit eingebauter Happy-Erweiterung.

Happy Compacter:

Nach dem Nachladen stößt der Anwender wieder in ein Menü, in dem es ihm möglich ist, mehrere Boot-Disketten auf einer zu sammeln. Tracks, die ein Programm nicht nutzt, werden für ein anderes freigehalten.

Multi Drive:

Bei diesem Kopierprogramm werden zwei Happy-Enhancement-Laufwerke benötigt. Dadurch entfällt das nervende dreimalige Auswechseln von Quell- und Zieldisketten.

Enable Tracer:

Der Tracer wird hier aktiviert. Jeden weiteren Track, der beim Diskettenzugriff angesprochen wird, merkt sich das Happy Drive Enhancement, und dieser kann nach erneutem Laden des »Happy Warp Speed Menü« abgefragt werden. Die Diskettenstation darf zwischendurch nicht ausgeschaltet werden, da sonst kein Tracen möglich wäre.

Display Trace:

Nur möglich, wenn vorher die Funktion »Enable Tracer« angewählt wurde. Es werden die Tracks 0 bis 39 angezeigt, die beim Booten benutzt wurden. Da beim Booten immer der Sektor 1 (Track 0) gelesen wird, und das Happy Warp Speed Menü nur Track 0 benutzt, ist das Aufzählen der gelesenen Tracks dem der vorherigen Boot-Diskette gleich. Deshalb ist das Happy Warp Speed Menü als Boot-Programm realisiert, als File könnte es auch noch andere Tracks belegen und so das Ergebnis verfälschen.

Insgesamt hinterläßt diese erste und vielseitigste Erweiterung einen hervorragenden Eindruck. Denn obwohl es andere und kostengünstigere Zusätze ähnlicher Art gibt, ist Happy Drive Enhancement das ausgereifteste Produkt. Außerdem unterstützen einige amerikanische und deutsche Softwarehersteller diesen Speeder mit ihren Programmen.

Mit diesen Programmen ist es möglich, sich ein völlig eigenes Format mit eigenem Kopierschutz zu erstellen. Mit der Atari 1050 kann man dann zum Beispiel ein Format in Medium-Density erzeugen, das jede 1050 ohne Probleme liest. Aber noch nicht einmal das Happy Drive könnte diese Diskette kopieren.

Das Happy-Drive-Enhancement ist für 149,95 Dollar plus 10 Dollar Versandkosten bei der Firma Happy-Computers, P.O. Box 1268, Morgan Hill, CA 95037 erhältlich.

Wer sich vor einer Bestellung in den USA scheut, kann sich an einen Atari-Händler wenden, muß dann aber mehr Geld ausgeben. Wie Sie eine Bestellung direkt in die USA richten, haben wir ausführlich in der Happy-Computer Ausgabe 7/87 beschrieben.

(Kjersten Waldheim/jg)

SPIELE-SPASS IST ANGESAGT!



Sonderheft 03/85: Spiele-Tests



Sonderheft 11: Spiele-Tests



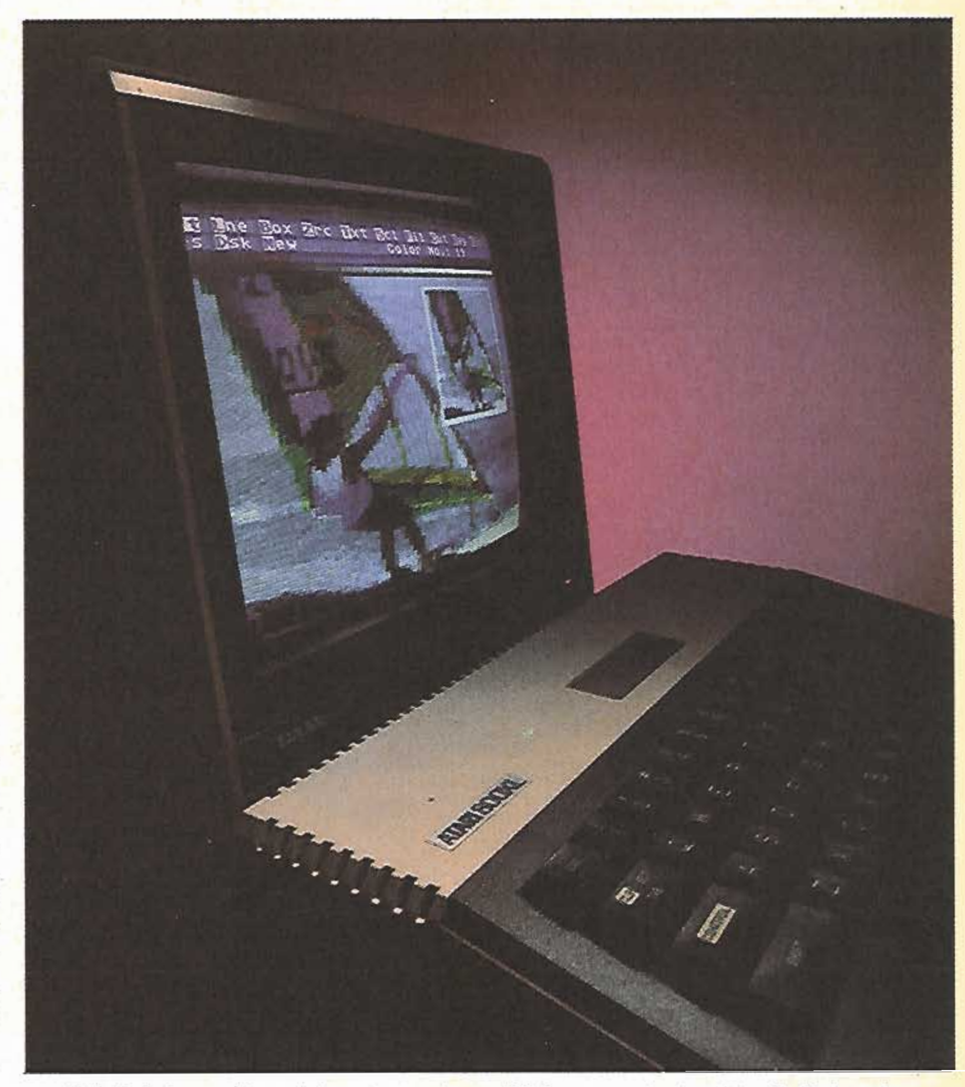
Sonderheft 17: Spiele-Tests

Achtung Spiele-Fans: Hier kommen insgesamt mehr als dreihundert Seiten voll mit Spiele-Tests, Marktübersichten, Grafik- und Musikprogrammen. Zusammen ergeben Sie ein Super-Nachschlagewerk: Geschicklichkeits-Spiele, Action-Spiele, Strategiespiele, Sport-Spiele …
vom C64 bis zum PC.

Nutzen Sie die Bestellmöglichkeit der Happy-Computer-Sonderhefte mit der eingehefteten Zahlkarte nach der Seite 98 in dieser Ausgabe.

Der XL treibt's ganz schön bunt

Mit ein bißchen Computerkenntnis, ein paar Assembler-Zeilen und einer guten Idee lassen sich aus den XL/XE-Computern viele Farben gleichzeitig herauskitzeln.



Eine der Stärken des Atari XL-Computers ist seine Farbenvielfalt. Er kann 16 Farben in 8 verschiedenen Helligkeiten darstellen, was zusammen 128 verschiedene Farben ergibt. Angesichts dieser Farbenvielfalt ist man begeistert, doch nach kurzer Zeit kommt die Ernüchterung: 128 Farbschattierungen stehen zwar zur Auswahl, in den einzelnen Grafikstufen sind aber nur maximal 4, in den Textstufen maximal 5 Farben gleichzeitig darstellbar. Es gibt zwar die Möglichkeit, über einen Display-List-Interrupt in jeder Bildschirmzeile die Farben umzuschalten. In vertikaler Richtung sind so beliebig viele Farben darstellbar, trotzdem hat man in einer horizontalen Zeile immer noch nur 4 beziehungsweise 5 Farben zur Verfügung.

Beim Atari gibt es außerdem 3 weitere Grafikstufen, in denen ein Bildpunkt eine Zeile hoch und vier Punkte breit ist. In der Grafikstufe 9 sind 16 Graustufen einer Farbe darstellbar und in Grafikstufe 11 sind es 16 Farben einer Helligkeit. Bei Grafikstufe 10 sind 7 Farben frei definierbar.

Um 256 Farben gleichzeitig auf dem Bildschirm darstellen zu können, ist folgende Überlegung notwendig: Das Fernseh- oder Monitorbild wird 50mal in der Sekunde neu gezeichnet. Wenn jetzt zwischen einem Bild mit Grauwerten und einem Bild mit Farbwerten im Takt des Bildschirmaufbaus hin- und hergeschaltet wird, kann das Auge des Betrachters die Bilder nicht mehr auseinanderhalten und mischt die Grau- und die Farbwerte zu einer Farbe zusammen. Auf diese Weise lassen sich aus 16 Grauwerten und 16 Farbwerten 256 Farben zurechtmischen.

Das Verfahren hat jedoch einige Nachteile, die hier nicht verschwiegen werden sollen. Zum ersten tritt durch das rasche Hin- und Herschalten zwischen den beiden Bildern ein unangenehmer Nebeneffekt auf: Das Bild flackert sehr stark. Dieser Effekt trat auch bei einigen alten Adventurespielen von Scott Adams auf. Nach den ersten drei Spielen sah er jedoch ein, daß diese Art der Bilderzeugung viel zu anstrengend für den Betrachter ist. Ab dem vierten Spiel verwendete er dann wieder die normale Art der Bilderzeugung. Ohne Bedeutung ist das Flackern jedoch, wenn man Bilder zeichnet und diese dann vom Bildschirm abfotografiert (siehe Bilder in diesem Beitrag und in der Inhaltsangabe dieser Ausgabe). Zum zweiten lassen sich bei diesem Verfahren, bedingt durch die Auflösung der beiden verwendeten Grafikstufen, horizontal nur 80 Punkte darstellen. Und zum dritten muß man einen nicht unerheblichen programmtechnischen Aufwand betreiben, um das Bild darzustellen. Im Computer muß nämlich ein Programm fest installiert sein, das die Bilder umschaltet. Zusätzlich nehmen diese Programmroutinen sowie die beiden Bilder Speicherplatz weg.

Dem Flackern läßt sich mit einem Kniff beikommen: Wie bei einem Fernsehbild verschiebt man jedes zweite übertragene Bild gegenüber dem ersten um eine Zeile. In jeder geraden Zeile werden dann die Grauwerte dargestellt und in jeder ungeraden Zeile die Farbwerte.

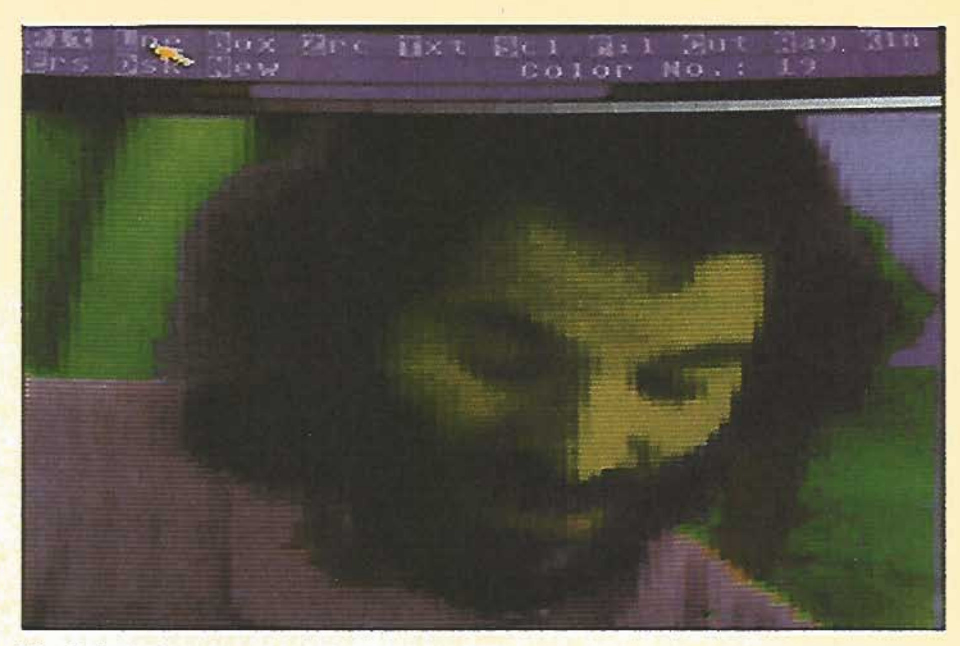
Damit man ein 256-Farben-Bild ausgeben kann, braucht man ein Programm, daß die Bilder hin- und herschaltet, sowie zwischen Grau- und Farbwerten wechselt. Zusätzlich braucht man zwei Display Lists, die das Farbbild gegenüber dem Helligkeitsbild um eine Zeile verschieben. Das Listing »COL256.BIN« stellt dieses Programm dar. Das Programm steht im Speicher ab Adresse 1536

($600) und belegt 104 Byte. Es ist so geschrieben, daß es unter Turbo-Basic mit dem Befehl »BLOAD "D:COL256.BIN"« geladen wird. Das Programm »FARB256« enthält alle Programmschritte, die nötig sind, um einen 256-Farben-Bildschirm zu erzeugen. Am besten speichern Sie das Programm mit »LIST "D:FARB 256.TXT"« auf Ihre Diskette, damit Sie es mit dem Befehl »ENTER« in eigene Programme einbinden können. Das Programm springt man unter Turbo-Basic einfach mit »EXEC FARB256_INIT« an, worauf die Display-Lists installiert und das COL256.BIN-Programm geladen werden.

Dann schaltet man die Grafik mit folgendem Befehl an:

```
D=USR($0600,GTIAMode)
```

Für die Variable »D« läßt sich jede andere im Programm nicht benützte Variable einsetzen. Sie wird von der Funktion zwar nicht verwendet, muß aber trotzdem angegeben werden, da der »USR«-Befehl eine Variable erwartet. Solange keine Player-Missile-Grafik verwendet wird, setzt man für GTIAMode normalerweise eine 64 ein, was Grafikmode 9 entspricht. Ansonsten schauen Sie bitte im Beitrag über die Hardwareregister der Atari 8-Bit-Computer unter dem Register



Mit »Paint256« erzeugtes, digitalisiertes Bild

»PRIOR« nach, welche Werte Sie einsetzen müssen. Nachdem die Grafik nun aktiviert ist, muß der Bildschirmspeicher gegen unbeabsichtigtes Überschreiben durch ein Basic-Programm geschützt werden. Dazu verwendet man folgende Befehlsfolge:

```
POKE 106,120
```

Will man jetzt Grafiken zeichnen, so muß die gleiche Grafik in beiden Bildern gezeichnet werden, einmal mit dem Helligkeitsbild und das zweite Mal mit dem Farbbild. Dazu legt man erst einmal die Farbe und die Helligkeit fest. So ergibt:

```
COL = 3
LUM = 6
```




ein dunkles Rot. Dann teilt man dem Computer über die Speicherstelle 88 mit, wo sein Bildschirmspeicher steht. Alle Grafikbefehle beziehen sich jetzt auf diesen Bildschirmspeicher. Als erstes zeichnet man das Helligkeitsbild.

```
DPOKE 88,AH
POKE 87,9
```

Anschließend setzt man den Helligkeitswert und führt den Grafikbefehl aus.

```
COLOR LUM
PLOT 0;0:DRAWTO 79,191
```

Dann setzt man den Punkt. Die gleiche Prozedur wiederholt man mit dem Farbwert.

```
DPOKE 88,AF
COLOR COL
PLOT 0,0:DRAWTO 79,191
```

Auf diese Weise lassen sich in jedes Programm Grafiken mit 256 Farben einbauen. Da unter Basic mit installierter Grafik nur noch rund 16 KByte Speicher zur Verfügung stehen, sollte man Programme unbedingt compilieren. Dadurch gewinnt man zusätzliche 4 KByte Speicher. Bevor man in irgendeiner Weise den seriellen Port benutzt, sollte man die Grafik wieder abschalten. Dies geschieht mit dem Befehl:

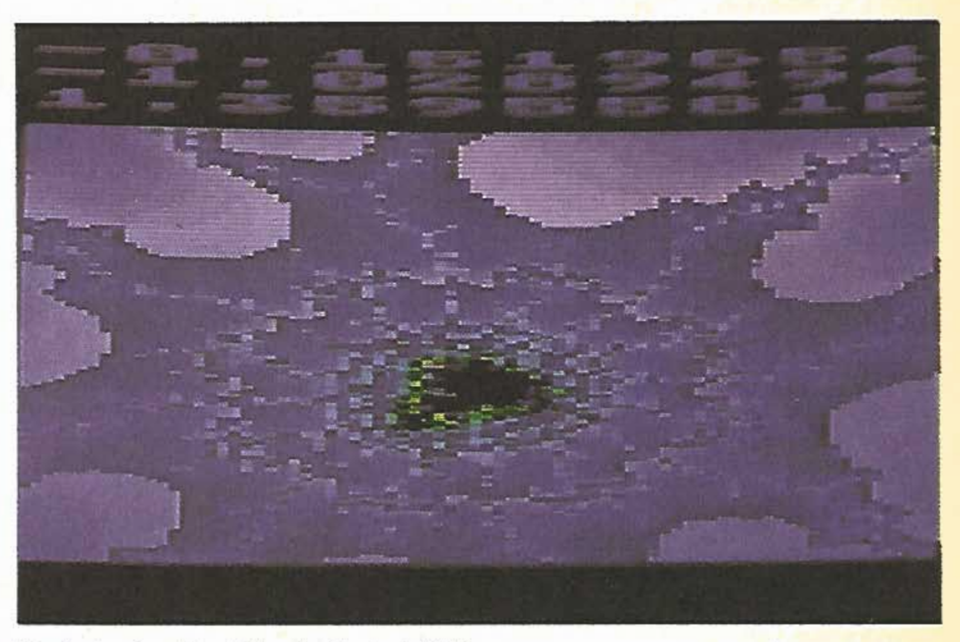
```
D=USR($0600,0)
```

Ansonsten hat der Computer nicht mehr genügend Rechenzeit zur Verfügung und hängt sich auf. Als Beispiel haben wir auf Seite 35 ein Programm abgedruckt, das alle 256 Farben gleichzeitig auf den Bildschirm bringt. Davon läßt sich auch sehr schön der Aufbau eines 256-Farben-Programms zeigen.

Nach der Arbeit kommt das Vergnügen

Zusätzlich zu den Routinen, die die 256 Farben ermöglichen, haben wir zwei Anwenderprogramme abgedruckt, die diese Routinen benutzen. Zum einen haben wir ein Malprogramm beigefügt. Es ist in Turbo-Basic geschrieben und benötigt die beiden Programme COL256.BIN sowie »DLST.256«. Diese Programme müssen sich zusammen mit »PAINT256« auf der gleichen Diskette befinden.

PAINT256 stellt verschiedene Funktionen zur Verfügung. Diese präsentieren sich nach dem Starten des Programms in der obersten Bildschirmzeile. Darunter folgen zwei schmale Zeilen mit 16 Farb- und 16 Helligkeitswerten, und unter diesen befindet sich die Zeichenfläche, auf der horizontal 80 und vertikal 192 Punkte darstellbar sind. Mit dem Pfeil, den man mit einem Joystick in Port 1 des Computers bewegt, werden dann die einzelnen Funktionen ausgewählt. Um eine Farbe festzulegen, fährt man mit dem Pfeil auf den entsprechenden Farb- oder Helligkeitsbalken und drückt den Feuerknopf. Zusätzlich läßt sich die Farbe mit den Pfeiltasten auf der Tastatur bestimmen. Die Aufwärts-/Abwärts-Pfeile verändern die Farbe und die Links-/Rechts-Pfeile bestimmen die Helligkeit einer Farbe. Die Nummer der ausgewählten Farbe wird in der zweiten Menüzeile über der Zeichenfläche bei »Color No.:« angezeigt.



Kunterbuntes Mandelbrot mit »Apfel256«

Mit dem Pfeil wählt man auch die einzelnen Zeichenfunktionen aus. Eine gewählte Funktion wird dann vollständig invers dargestellt. Folgende Funktionen stehen zur Verfügung, wobei sich fast alle Funktionen mit <ESC> abbrechen lassen, wenn sie einmal versehentlich angewählt wurden.

Plt (Plot): Wenn der Knopf gedrückt ist, setzt das Programm an der entsprechenden Stelle auf dem Bildschirm einen Punkt.

Lne (Line) zieht eine Linie von dem Punkt, an dem das erstemal der Knopf betätigt wurde, zu dem Punkt des zweiten Knopfdrucks.

Box (Box): Der erste Knopfdruck legt den Eckpunkt eines Kastens fest. Wird dann der Pfeil bewegt, so wird von dem Eckpunkt zu der neuen Pfeilposition ein Kasten gezogen. Beim zweiten Knopfdruck wird der Kasten gezeichnet.

Zrc (Circle): Beim ersten Knopfdruck wird der Mittelpunkt eines Kreises bestimmt. Beim zweiten wird ein Kreis mit dem Radius vom Abstand des Mittelpunkts zur Position des Pfeils gezeichnet.

Txt (Text): Nach dem der Knopf betätigt wurde, läßt sich ein Text mit maximal 10 Zeichen Länge eingeben. Mit <BACKSPACE> wird eine fehlerhafte Eingabe korrigiert. <ESC> löscht den gesamten Text. Mit <RETURN> wird der Text übernommen und mit dem Joystick positioniert. Durch Knopfdruck wird der Text in das Bild eingefügt.

Scl (Scale): Mit dieser Funktion wird der Ausschnitt eines Bildes vergrößert oder verkleinert. Dabei dürfen sich die Bereiche auch überlappen. Da der Computer sehr viel Rechenzeit verbraucht, wenn er einen Bildschirmausschnitt transformiert, wird zwischendurch der Bildschirm ausgeschaltet. Durch diesen Trick hat der Computer mehr Zeit zur Verfügung.

Fil (Fill): Eine schwarze Fläche wird mit der gewählten Farbe ausgefüllt.

Cut (Cut): Diese Funktion schneidet einen rechteckigen Ausschnitt aus und fügt ihn an anderer Stelle wieder ein.

Ray (Rays): Ausgehend von einem gemeinsamen Punkt werden Linien gezogen. Indem man im Menü eine andere Funktion auswählt, beendet man diese Funktion.

Cln (continue Lines): Es werden fortlaufende Linien gezogen. Der Endpunkt einer Linie ist gleichzeitig der Startpunkt der nächsten Linie. Auch diese Funktion wird erst durch eine andere Funktion beendet.

Ers (Erase): Der Pfeil verwandelt sich in ein schraffiertes Zeichen, das eine Art Radiergummi darstellt. Es löscht Flächen aus einem Bild.



Superbase – das relationale Datenbank-System für den Atari ST

Superbase vereint als erstes Programm einer neuen Generation von Datenbank-Systemen sowohl eine neuartige, äußerst benutzerfreundliche Bedienung mit Pull-down-Menüs, Fenstern und Maussteuerung als auch die enorme Leistungsfähigkeit einer relationalen Dateiverwaltung.

Einfacher Datenbank-Aufbau

Mit den leichtverständlichen Pull-down-Menüs und Kontrollfeldern legen Sie in Minuten eine komplette Datenbank an. Sie können eine bereits festgelegte Struktur jederzeit ändern, ohne Ihre Daten zu zerstören.

Verwaltung der Daten

Superbase zeigt Ihre Daten auf verschiedene Arten an, beispielsweise als Tabelle oder als Formular. Sind Index und Felder selektiert, können Sie Ihre Daten wie bei einem Videorecorder anzeigen lassen. Schneller Vorlauf, Rücklauf, Pause und Stop – ein Recorder ist nicht einfacher zu bedienen. Ein einzigartiges Filtersystem wählt beliebige Datenkategorien aus, mit denen Sie dann arbeiten können.

Die Stärken von Superbase

Das Festlegen von Übersichten und zusammenhängende Abfragen über mehrere verknüpfte Dateien sind auch bei verschiedenen Sortierkriterien kein Problem. Daten anderer Datenbanken oder Anwenderprogramme im ASCII-Format lassen sich ebenfalls problemlos verarbeiten. Binden Sie Daten in Ihre Textverarbeitung ein, oder bilden Sie aus verschiedenen Dateien eine neue Datenbank!

Datenbank mit Bildern

Superbase bietet neben den gängigen Datenbank-Funktionen die Möglichkeit, Bilder und Grafiken darzustellen und zu verwalten. Einzigartigen Grafik-Datenbanken oder Dia-Shows steht also nichts im Wege.

Wer braucht Superbase?

Die Anwendungsmöglichkeiten sind nahezu unbegrenzt.

Hier einige Beispiele:

Geschäftliches	Professionelle *Anwendungen*
Lagerbestand Fakturierung Registratur Versandlisten Verwaltung Adressen	Design Fotografie Journalismus Sammlungen Forschung Ausbildung

Leistungsumfang

Die Software: ● bis zu 16 Millionen Datensätze pro Datei ● maximal 999 Indizes pro Datei ● Speicherkapazität pro Datei, Anzahl der geöffneten Dateien, Anzahl der Dateien und Anzahl der Felder pro Datensatz: abhängig von der Disketten- bzw. Festplattenkapazität.

Die Daten: ● Text, Daten, numerische Felder und externe Dateien ● Überprüfung bei der Eingabe ● Formelfelder ● Kalender der Jahre 1–9999, verschiedene Datumsformen ● verschiedene Zahlenformate bei 13stelliger Genauigkeit ● Datenschutz per Paßwort

Die Ausgaben: ● das Programm beherrscht einen flexiblen Etikettendruck und produziert übersichtliche Listen mit dem Reportgenerator ● bis zu 255 Spalten ● mit Titel, Datum und Seitenzahl ● Datensatz-Zähler, Durchschnitt, Zwischen- und Endergebnis ● Ausgabe von mehreren Dateien auf Bildschirm, Drucker, Diskette oder neuer Datei ● mehrspaltiger Etikettendruck mit variablem Format ● Speicherung der Ausgabe- und Abfrage-Formate zur späteren Verwendung ● vielfältige Sortierkriterien

Hardware-Anforderung

● Atari ST 512 Kbyte ● 3½"-Diskettenlaufwerk ● 80-Zeichen-Monitor

Lieferumfang: ● 3½"-Diskette ● Handbuch deutsch

Best.-Nr. 51641

DM 249,–* (sFr 199,–/öS 2490,–*)

* inkl. MwSt. Unverbindliche Preisempfehlung

Übrigens: Superbase gibt es auch für Amiga, Schneider PC und IBM-PCs und Kompatible



Markt&Technik

Zeitschriften · Bücher

Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

705216

Mandelbrot-Ausschnitte			
Realteil	Imaginärteil	Seitenlänge	Ausschnitt
-2	-1.25	2.5	Gesamtansicht
-0.225	-1.125	0.25	Zweig
-0.74591	0.11196	1.43E-3	Spirale
-0.1765625	-1.0484375	0.03125	kleines Mandelbrot

Dsk (Disk): Es erscheint ein neues Menü, über das verschiedene Diskettenfunktionen ausgeführt werden:

- **Save** (Save): Das Bild wird auf der Diskette gespeichert, wobei ein Bild 123 Sektoren belegt.
- **Load** (Load): Sämtliche Files mit dem Zusatz ».PIC« werden in der obersten Zeile angezeigt. Ein Knopfdruck lädt das Bild. Indem man den Joystick bewegt, wird das nächste File angezeigt.
- **Frmt** (Format): Formatiert die Diskette in Laufwerk 1. Dazu muß die Taste <Y> gedrückt werden. Alle anderen Tasten brechen die Funktion sofort ab.
- **Dir** (Directory): Zeigt sämtliche Files mit dem Zusatz ».PIC« an. Durch Joystickbewegung wird das nächste File aufgeführt, Knopfdruck beendet die Funktion.
- **EDir** (extended Directory): Dies funktioniert wie die Directory-Funktion. Im Gegensatz dazu werden jedoch alle Files ausgegeben.
- **Bye** (Bye): Das Programm kehrt wieder ins Hauptmenü zurück.
- **Quit** (Quit): Das Programm wird beendet, und man befindet sich wieder im Turbo-Basic.

New (New): Nach Drücken der Taste <Y> wird das Bild gelöscht.

Der Regenbogen über dem Mandelbrot

Das zweite Programm berechnet die gute alte Mandelbrotmenge, bei dieser Version allerdings in 256 Farben. Das Prinzip der Mandelbrotberechnung soll hier nicht weiter erläutert werden. Gerade in der letzten Zeit haben sich wissenschaftliche Magazine und Computerzeitschriften mit Beiträgen über die Mandelbrotmenge geradezu überschlagen, so daß die Berechnung der Bilder bekannt sein sollte. Wer trotzdem wissen möchte, wie Mandelbrot funktioniert, und sich allgemein für Fractals interessiert, sollte sich das Buch »The Beauty of Fractals« von H.-O. Pleitgen und P. H. Richter anschauen. Es ist zwar in englischer Sprache verfaßt, bietet aber soviel Stoff zu diesem Thema, daß es fast als »Bibel« der Mandelbrotmenge und deren Nebenarten bezeichnet werden kann. Nebenbei enthält es viele phantastische Bilder. Das Buch ist im Springer-Verlag Heidelberg erschienen und kostet 78 Mark.

Zurück zum Mandelbrotprogramm. Gerade bei Bildern aus der Mandelbrotmenge sind viele Farben wichtig, da die Bilder sonst einen Großteil ihres Reizes verlieren. Auch dieses Programm ist in Turbo-Basic geschrieben und sollte mit dem Prüfsummer eingegeben werden. Nach dem Starten meldet sich das Programm mit folgendem Menü:

Laden Bild: Das Programm verlangt eine dreistellige Nummer des zu ladenden Bildes. Anschließend wird das Bild mit dem Dateinamen »APFELXXX.PIC« geladen.

Speicher Bild: Wie bei »Laden Bild«, nur daß bei dieser das Bild vom Speicher auf Diskette übertragen wird.

Berechne Bild: Nach Eingabe der Werte für die untere linke Ecke, der Seitenlänge und der Berechnungstiefe beginnt das Programm einen Ausschnitt aus der Mandelbrotmenge zu berechnen. Dieser Ausschnitt ist wie bei den Originalprogrammen auf Großcomputern quadratisch. Die Berechnungstiefe sollte größer oder gleich 256 sein, da sonst nicht alle darstellbaren Farben ausgenutzt werden.

Ausschnitt vom Bild: Diese Funktion läßt sich nur aufrufen, wenn sich ein Bild im Speicher des Computers befindet. Das Programm schaltet auf das Bild um, und es erscheinen zwei Rahmen auf dem Bildschirm, die die rechte obere sowie die linke untere Ecke eines neu zu berechnenden Bildausschnitts repräsentieren. Der Ausschnitt wird mit dem Joystick über das Bild bewegt. Ein Druck auf den Joystickknopf verändert die Ausschnittsgröße. Bei Druck auf die <START>-Taste fragt das Programm nach der Rechentiefe. Anschließend wird der neue Mandelbrotausschnitt berechnet.

Während die Berechnung des Bildes erfolgt, läßt sich der Bildschirm mit der <START>-Taste abschalten. Dadurch arbeitet der Computer doppelt bis dreimal so schnell. Mit der <SELECT>-Taste wird der Bildschirm wieder angeschaltet.

Directory: Das Inhaltsverzeichnis des Diskettenlaufwerks wird angezeigt.

<SELECT>-Taste: Das berechnete Bild erscheint. Falls kein Bild im Speicher ist, bleibt der Bildschirm schwarz.

<START>-Taste: Es wird wieder zum Menü zurückgeschaltet.

Im unteren Bereich des Bildschirms werden die Bildparameter des berechneten beziehungsweise geladenen Bildes angezeigt. Ein Bild nimmt auf der Diskette zwischen 123 und 124 Sektoren in Anspruch, je nachdem, wie viele Stellen die Bildwerte aufweisen. Damit Sie, ohne viel in der Mandelbrotmenge herumzusuchen, Bilder mit dem Programm »APFEL256« berechnen können, haben wir für Sie Werte interessanter Ausschnitte zusammengestellt.

Vielleicht wissen Sie weitere interessante Anwendungen für den 256-Farben-Zusatz. Schreiben Sie uns doch Ihre Vorschläge für weitere Anwendungen. Wir freuen uns über jede Anregung.

(Stephan Gerle/hf)

Steckbrief	
Programm:	Paint256
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette

Steckbrief	
Programm:	Apfel256
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette

```
Programmname :COL256.BIN
Länge        :112 Bytes
------------------------------------

0000:FF FF 00 06 67 06 68 68 <9A>
0008:68 C9 00 F0 20 85 D0 A9 <C9>
0010:FF 8D 66 06 A2 06 A0 36 <B2>
0018:A9 07 20 5C E4 A9 57 8D <7D>
0020:00 02 A9 06 8D 01 02 A9 <4B>
0028:C0 8D 0E D4 60 A2 E4 A0 <5F>
0030:62 A9 07 20 5C E4 A9 60 <8A>
0038:8D 0E D4 60 A5 D0 29 7F <10>
0040:85 D0 A2 00 A0 7C AD 66 <9D>
0048:06 49 80 8D 66 06 30 04 <FD>
0050:A2 00 A0 78 8E 30 02 8C <79>
0058:31 02 4C 62 E4 48 A5 D0 <6A>
0060:49 80 85 D0 8D 0A D4 8D <1B>
0068:1B D0 68 40 00 00 00 00 <2F>
```

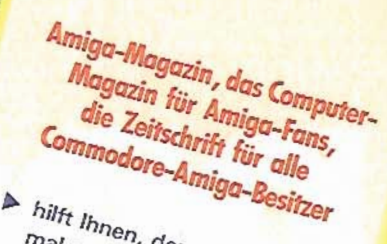
Listing. »COL256.BIN«

```
Programmname :DLST.256
Länge        :1192 Bytes
-----------------------------------
0000:70 00 00 42 10 9E 00 02 <59>
0008:90 0F 0F 0F 0F 90 0F 0F <CA>
0010:0F 0F 10 CF 10 80 CF 38 <9E>
0018:A0 CF 60 80 CF 88 A0 CF <AC>
0020:B0 80 CF D8 A0 CF 00 81 <7C>
0028:CF 28 A1 CF 50 81 CF 78 <38>
0030:A1 CF A0 81 CF C8 A1 CF <20>
0038:F0 81 CF 18 A2 CF 40 82 <F8>
0040:CF 68 A2 CF 90 82 CF B8 <A2>
0048:A2 CF E0 82 CF 08 A3 CF <93>
0050:30 83 CF 58 A3 CF 80 83 <44>
0058:CF A8 A3 CF D0 83 CF F8 <2D>
0060:A3 CF 20 84 CF 48 A4 CF <4F>
0068:70 84 CF 98 A4 CF C0 84 <D0>
0070:CF E8 A4 CF 10 85 CF 38 <9B>
0078:A5 CF 60 85 CF 88 A5 CF <A3>
0080:B0 85 CF D8 A5 CF 00 86 <7F>
0088:CF 28 A6 CF 50 86 CF 78 <CA>
0090:A6 CF A0 86 CF C8 A6 CF <93>
0098:F0 86 CF 18 A7 CF 40 87 <03>
00A0:CF 68 A7 CF 90 87 CF B8 <F4>
00A8:A7 CF E0 87 CF 08 A8 CF <49>
00B0:30 88 CF 58 A8 CF 80 88 <53>
00B8:CF A8 A8 CF D0 88 CF F8 <3E>
00C0:A8 CF 20 89 CF 48 A9 CF <8A>
00C8:70 89 CF 98 A9 CF C0 89 <DF>
00D0:CF E8 A9 CF 10 8A CF 38 <B4>
00D8:AA CF 60 8A CF 88 AA CF <C2>
00E0:B0 8A CF D8 AA CF 00 8B <8E>
00E8:CF 28 AB CF 50 8B CF 78 <5E>
00F0:AB CF A0 8B CF C8 AB CF <92>
00F8:F0 8B CF 18 AC CF 40 8C <96>
0100:CF 68 AC CF 90 8C CF B8 <46>
0108:AC CF E0 8C CF 08 AD CF <9F>
0110:30 8D CF 58 AD CF 80 8D <D9>
0118:CF A8 AD CF D0 8D CF F8 <88>
0120:AD CF 20 8E CF 48 AE CF <5B>
0128:70 8E CF 98 AE CF C0 8E <F7>
0130:CF E8 AE CF 10 8F CF 38 <C6>
0138:AF CF 60 8F CF 88 AF CF <0F>
0140:B0 8F CF D8 AF CF 00 90 <96>
0148:CF 28 B0 CF 50 90 CF 78 <81>
0150:B0 CF A0 90 CF C8 B0 CF <A0>
0158:F0 90 CF 18 B1 CF 40 91 <AC>
0160:CF 68 B1 CF 90 91 CF B8 <1B>
0168:B1 CF E0 91 CF 08 B2 CF <96>
0170:30 92 CF 58 B2 CF 80 92 <69>
0178:CF A8 B2 CF D0 92 CF F8 <5D>
0180:B2 CF 20 93 CF 48 B3 CF <96>
0188:70 93 CF 98 B3 CF C0 93 <F2>
0190:CF E8 B3 CF 10 94 CF 38 <58>
0198:B4 CF 60 94 CF 88 B4 CF <CE>
01A0:B0 94 CF D8 B4 CF 00 95 <A5>
01A8:CF 28 B5 CF 50 95 CF 78 <92>
01B0:B5 CF A0 95 CF C8 B5 CF <00>
01B8:F0 95 CF 18 B6 CF 40 96 <32>
01C0:CF 68 B6 CF 90 96 CF B8 <F4>
01C8:B6 CF E0 96 CF 08 B7 CF <91>
01D0:30 97 CF 58 B7 CF 80 97 <78>
01D8:CF A8 B7 CF D0 97 CF F8 <A6>
01E0:B7 CF 20 98 CF 48 B8 CF <55>
01E8:70 98 CF 98 B8 CF C0 98 <FC>
01F0:CF E8 B8 CF 10 99 CF 38 <F4>
01F8:B9 CF 60 99 CF 88 B9 CF <7D>
0200:B0 99 CF D8 B9 CF 00 9A <9D>
0208:CF 28 BA CF 50 9A CF 78 <24>
0210:BA CF A0 9A CF C8 BA CF <AC>
0218:F0 9A CF 18 BB CF 40 9B <42>
0220:CF 68 BB CF 90 9B CF B8 <4E>
0228:BB CF E0 9B CF 08 BC CF <00>
0230:30 9C CF 58 BC CF 80 9C <7A>
0238:CF A8 BC CF D0 9C CF F8 <78>
0240:BC CF 20 9D CF 48 BD CF <83>
0248:70 9D CF 98 BD CF C0 9D <10>
0250:CF E8 BD 41 70 00 00 42 <16>
0258:10 9E 00 02 90 0F 0F 0F <D1>
0260:0F 90 0F 0F 0F 0F 90 CF <45>
0268:10 A0 CF 38 80 CF 60 A0 <59>
0270:CF 88 80 CF B0 A0 CF D8 <50>
0278:80 CF 00 A1 CF 28 81 CF <E1>
0280:50 A1 CF 78 81 CF A0 A1 <45>
0288:CF C8 81 CF F0 A1 CF 18 <3B>
0290:82 CF 40 A2 CF 68 82 CF <A5>
0298:90 A2 CF B8 82 CF E0 A2 <53>
02A0:CF 08 83 CF 30 A3 CF 58 <71>
02A8:83 CF 80 A3 CF A8 83 CF <55>
02B0:D0 A3 CF F8 83 CF 20 A4 <F2>
02B8:CF 48 84 CF 70 A4 CF 98 <DB>
02C0:84 CF C0 A4 CF E8 84 CF <AA>
02C8:10 A5 CF 38 85 CF 60 A5 <68>
02D0:CF 88 85 CF B0 A5 CF D8 <25>
02D8:85 CF 00 A6 CF 28 86 CF <2C>
02E0:50 A6 CF 78 86 CF A0 A6 <E4>
02E8:CF C8 86 CF F0 A6 CF 18 <D1>
02F0:87 CF 40 A7 CF 68 87 CF <64>
02F8:90 A7 CF B8 87 CF E0 A7 <DE>
0300:CF 08 88 CF 30 A8 CF 58 <F3>
0308:88 CF 80 A8 CF A8 88 CF <8B>
0310:D0 A8 CF F8 88 CF 20 A9 <2A>
0318:CF 48 89 CF 70 A9 CF 98 <ED>
0320:89 CF C0 A9 CF E8 89 CF <1F>
0328:10 AA CF 38 8A CF 60 AA <0E>
0330:CF 88 8A CF B0 AA CF D8 <89>
0338:8A CF 00 AB CF 28 8B CF <9E>
0340:50 AB CF 78 8B CF A0 AB <FE>
0348:CF C8 8B CF F0 AB CF 18 <E7>
0350:8C CF 40 AC CF 68 8C CF <92>
0358:90 AC CF B8 8C CF E0 AC <69>
0360:CF 08 8D CF 30 AD CF 58 <88>
0368:8D CF 80 AD CF A8 8D CF <42>
0370:D0 AD CF F8 8D CF 20 AE <78>
0378:CF 48 8E CF 70 AE CF 98 <FE>
0380:8E CF C0 AE CF E8 8E CF <76>
0388:10 AF CF 38 8F CF 60 AF <E1>
0390:CF 88 8F CF B0 AF CF D8 <01>
0398:8F CF 00 B0 CF 28 90 CF <18>
03A0:50 B0 CF 78 90 CF A0 B0 <05>
03A8:CF C8 90 CF F0 B0 CF 18 <FA>
03B0:91 CF 40 B1 CF 68 91 CF <50>
03B8:90 B1 CF B8 91 CF E0 B1 <82>
03C0:CF 08 92 CF 30 B2 CF 58 <99>
03C8:92 CF 80 B2 CF A8 92 CF <81>
03D0:D0 B2 CF F8 92 CF 20 B3 <3F>
03D8:CF 48 93 CF 70 B3 CF 98 <13>
03E0:93 CF C0 B3 CF E8 93 CF <31>
03E8:10 B4 CF 38 94 CF 60 B4 <A5>
03F0:CF 88 94 CF B0 B4 CF D8 <93>
03F8:94 CF 00 B5 CF 28 95 CF <1F>
0400:50 B5 CF 78 95 CF A0 B5 <94>
0408:CF C8 95 CF F0 B5 CF 18 <95>
0410:96 CF 40 B6 CF 68 96 CF <8E>
0418:90 B6 CF B8 96 CF E0 B6 <90>
0420:CF 08 97 CF 30 B7 CF 58 <A3>
0428:97 CF 80 B7 CF A8 97 CF <3E>
0430:D0 B7 CF F8 97 CF 20 B8 <0F>
0438:CF 48 98 CF 70 B8 CF 98 <35>
0440:98 CF C0 B8 CF E8 98 CF <6F>
0448:10 B9 CF 38 99 CF 60 B9 <5F>
0450:CF 88 99 CF B0 B9 CF D8 <A1>
0458:99 CF 00 BA CF 28 9A CF <25>
0460:50 BA CF 78 9A CF A0 BA <1B>
0468:CF C8 9A CF F0 BA CF 18 <26>
0470:9B CF 40 BB CF 68 9B CF <75>
0478:90 BB CF B8 9B CF E0 BB <A7>
0480:CF 08 9C CF 30 BC CF 58 <3C>
0488:9C CF 80 BC CF A8 9C CF <6D>
0490:D0 BC CF F8 9C CF 20 BD <26>
0498:CF 48 9D CF 70 BD CF 98 <46>
04A0:9D CF C0 BD CF E8 9D 41 <8F>
04A8:00 00 00 00 00 00 00 00 <87>
```

Listing. »DLST.256«

```
10 --                                      <YW>
20 REM Paint 256  (c) by Stephan Gerle &
 Francois Paquet-Durand                    <PL>
30 REM Tel.: 02506/7493 (Stephan) ; 0252
6/2120 (Francois)                          <BF>
40 --                                      <YZ>
1020 EXEC DECLARE:EXEC MENUE_INIT:EXEC A
N                                          <PS>
1040 POKE 709,12:POKE 710,98               <RW>
1070 # START                               <NG>
1071 POP :TRAP #FEHLER                     <SB>
1072 POKE 87,%0:DPOKE 88,ME:PLOT %3,%3:P
OSITION 32,%1:? COL*16+LUM;"▲▲";           <EV>
2000 --                                    <VF>
2010 DO                                    <TQ>
2020 EXEC WAIT_KEY                         <JN>
2030 CHALT=CH                              <XG>
2040 IF T>%0                               <FL>
2050 T=T&127:D=%0                          <IL>
2060 IF T=42 THEN LUM=LUM+(LUM<15):D=%1    <KU>
2070 IF T=43 THEN LUM=LUM-(LUM>%0):D=%1    <UQ>
2080 IF T=61 THEN COL=COL+(COL<15):D=%1    <TK>
2090 IF T=45 THEN COL=COL-(COL>%0):D=%1    <DU>
2100 IF D=%1 THEN GO# START                <OR>
2110 T=T&95                                <OB>
2120 IF T=80 THEN CH=%1                    <RP>
2130 IF T=76 THEN CH=%2                    <UY>
2140 IF T=66 THEN CH=%3                    <VN>
2150 IF T=90 THEN CH=4                     <NS>
2160 IF T=84 THEN CH=5                     <QC>
2170 IF T=83 THEN CH=6                     <QP>
2180 IF T=70 THEN CH=7                     <PT>
2190 IF T=67 THEN CH=8                     <TN>
2200 IF T=82 THEN CH=9                     <RY>
2210 IF T=75 THEN CH=10                    <CS>
2211 IF T=69 THEN CH=11                    <FE>
2212 IF T=68 THEN CH=12                    <FT>
2213 IF T=78 THEN CH=13                    <HF>
2220 ELSE                                  <TF>
2230 IF Y<-23                              <KG>
2240 CH=X DIV 8+%1                         <XH>
2250 ELSE                                  <TO>
2251 IF Y<-14 AND X<22                     <JE>
2252 CH=10+X DIV 8+%1                      <MK>
2253 ELSE                                  <UA>
2260 IF Y<-8 AND Y>-12                     <GB>
2270 COL=X DIV 5:GO# START                 <AN>
2280 ELSE                                  <TX>
2290 IF Y<-2 AND Y>-7                      <SE>
2300 LUM=X DIV 5:GO# START                 <DU>
2310 ENDIF                                 <SQ>
2320 ENDIF                                 <ST>
2340 ENDIF                                 <SZ>
2341 ENDIF                                 <TD>
2350 ENDIF                                 <TC>
2360 IF CHALT<>CH                          <TA>
2370 POS=CHALT*4-%3+ME                     <LU>
2380 POKE POS,PEEK(POS)&127                <MK>
2390 POKE POS+%1,PEEK(POS+%1)&127          <IO>
2400 POS=CH*4-%3+ME                        <FV>
2410 POKE POS,PEEK(POS)!128                <IC>
2420 POKE POS+%1,PEEK(POS+%1)!128          <DI>
2430 ENDIF                                 <SY>
2440 IF Y>-%1 AND STRIG(%0)=%0 OR CH>11    <LR>
2450 EXEC SAFE:XA=X:YA=Y:EXEC HELL:D=%0    <HI>
2460 ON CH EXEC PLOT,LNE,BOX,CRC,TXT,SCL
,FIL,CUT,RAY,RAY,RUB,DSK,NEW               <JM>
2470 ENDIF                                 <TK>
```

Listing. »Paint256«



- hilft Ihnen, den Amiga maximal zu nutzen
- bringt für Einsteiger und Experten, Hobby- und Profiprogrammierer-Kurse in CLI, BASIC, MODULA II, »C« etc. etc.
- testet für Sie Hardware, Peripherie und aktuellste Software sämtlicher Hersteller
- anspruchsvolle Listings und Anwendungen geben Ihrer Arbeit höchste Effizienz
- in Kursen optimieren Sie die Bedienung Ihres Amiga

Kennenlern-Angebot

mit kostenlosem »Amiga«-Probeexemplar und Poster

Ja, ich möchte eine kostenlose Ausgabe von »Amiga-Magazin« zur Probe. Wenn ich »Amiga-Magazin« weiterlesen will, brauche ich nichts zu tun, ich erhalte dann »Amiga-Magazin« regelmäßig für mindestens 12 Ausgaben zum günstigen Preis von 79,– DM (Ausland 97,– DM). Das Abonnement verlängert sich automatisch um ein Jahr, wenn es vor Ablauf nicht gekündigt wird. Ich erhalte mit meinem Probeexemplar das »Amiga«-Poster, das ich in jedem Fall behalten kann.

Name, Vorname

Telefon

Straße

PLZ, Wohnort

Datum, 1. Unterschrift

Coupon ausschneiden und einsenden an:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München.

Ich weiß, daß ich diese Bestellung innerhalb von 8 Tagen widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs an Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München. Ich bestätige dies durch meine 2. Unterschrift.

Datum, 2. Unterschrift

HCS20

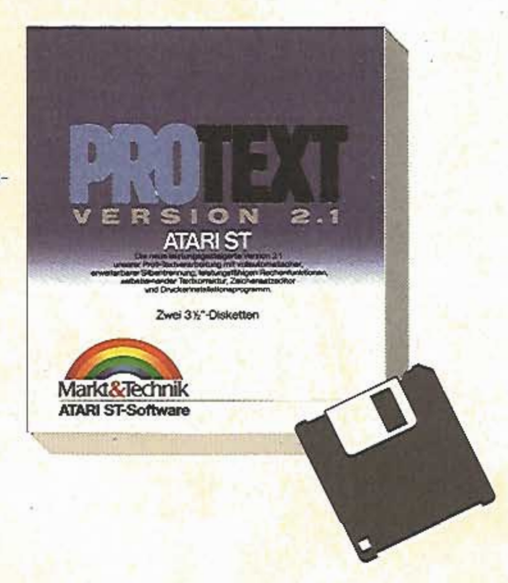
ATARI

PROTEXT für die ATARI ST

Version 2.1

Die professionelle deutsche Textverarbeitung mit vollautomatischer Silbentrennung und integrierter Tabellenkalkulation.

Ab sofort erhalten Sie PROTEXT in einer leistungsstärkeren Version – zum alten Preis. Als Besitzer der Version 1.x (1.0–1.3) haben Sie die Möglichkeit, kostengünstig ein Update zu erhalten.
PROTEXT ist ein leicht bedienbares, Maus-unterstütztes Textprogramm mit hoher Leistungsfähigkeit. Eingebaute Hilfefunktionen ermöglichen auch dem Anfänger, die gesamte Leistungsfähigkeit dieser professionellen Software zu nutzen. Das Programm erlaubt die direkte Eingabe und Änderung aller Attribute wie **Fettschrift**, Unterstreichen, *Kursivschrift*, Breitschrift, Hoch- und Tiefstellen. Der Text ist ohne besondere Ausgabe auf dem Bildschirm sofort formatiert sichtbar, so wie er auch auf dem Drucker ausgegeben wird. Der Zeichensatz ist frei definierbar. Alle Positionen im Text (vom verfügbaren Speicherplatz abhängig) können mit entsprechenden Funktionen sehr schnell aufgesucht werden.
Die schon bekannten Leistungsmerkmale: vollautomatische Silbentrennung, der Text wird auf dem Bildschirm genauso ausgegeben, wie er später ausgedruckt wird (WYSIWYG-Prinzip), komfortable Funktionsübersicht, universelle Druckeranpassung, Verketten von Texten, Archiv-Funktionen, komfortable, programmierbare Rechenfunktionen, selbstlernende Textkorrektur mit Grundwortschatz, Textlupe, Zeichensatzanzeige.



Neue Features:
Block markieren jetzt direkt mit Maus möglich, Maskenmodus, Zugriff auf im Desktop installierte Accessories, Text direkt drucken, Ganzseitenübersicht, erweiterbare, verbesserte Silbentrennung, optimierte Druckerinstallation, geschwindigkeitsgesteigerte Formatierroutine und vieles mehr.

Hardware-Anforderung:

- Atari 260 ST, 520 ST, 520 ST+, 1040 ST,
- Monochrommonitor
- beliebiger Drucker

Bestell-Nr. 51643

Für nur DM 148,–*

PROTEXT Update 1.x auf 2.1:

Bestell-Nr. 51644

Für nur DM 30,–*

(gegen Einsendung der Originaldiskette oder Registrierungskarte)

* inkl. MwSt. Unverbindliche Preisempfehlung

	Version	Best.-Nr.	Format	Preis DM	sFr	öS
WordStar/MailMerge	Atari ST	50106	3½"	199,-*	178,-	1990,-*
dBASE II	Atari ST	50306	3½"	348,-*	295,-	3490,-*
Update zu Protext 1.x	Atari ST	51644	3½"	30,-*	30,-	299,-*
Protext 2.1	Atari ST	51643	3½"	148,-*	132,-	1490,-*

* inkl. MwSt. Unverbindliche Preisempfehlung



708217

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

```
2480 POKE 764,255                          <GY>
2490 LOOP                                  <ZE>
2990 --                                    <WY>
3000 PROC DECLARE                          <DF>
3010 DIM LEER$(256),MAUS$(11),NAME$(16),
UNDO$(7680)                                <EK>
3031 PMB=120:POKE 54279,PMB:PMB=PMB*256+
1839                                       <EC>
3040 CH=%1:CHALT=CH:XX=%1:COL=%1:LUM=%3:
UN=ADR(UNDO$)                              <KW>
3090 POKE 707,255:POKE 53277,%2:MAUS$="{
CTL ,}{CTL ,}@p{SHIFT =}x<.{CTL G}{CTL B
}{CTL ,}":MOVE ADR(LEER$),PMB,256         <CF>
3120 AH=$8010:AF=$A010:ME=AH+$1E00         <XR>
3140 OPEN #%1,4,%0,"D:DLST.256":BGET #%1
,$7800,596:BGET #%1,$7C00,596:CLOSE #%1    <PU>
3170 BLOAD "D:COL256.BIN"                  <OW>
3180 POKE AF,%0:MOVE AF,AF+%1,$1E00:MOVE
 AF,AH,$1E00                               <GC>
3210 POKE 106,120:GRAPHICS %0:POKE 82,%0   <YV>
3290 ENDPROC                               <PJ>
3300 PROC PARTSAFE                         <OV>
3310 IF YA<Y:YY=YA*40:ELSE :YY=Y*40:ENDI
F :MOVE UN+YY,AH+YY,ABS(YA-Y)*40+40        <CA>
3390 ENDPROC                               <PL>
3530 PROC MENUE                            <CV>
3540 DPOKE 88,ME:PLOT %3,%3:POSITION %0,
%0:MOVE ADR(LEER$),ME,60:POKE 87,%0        <YR>
3570 ENDPROC                               <PJ>
3580 PROC HELL                             <HQ>
3590 DPOKE 88,AH:POKE 87,9:COLOR LUM       <EP>
3600 ENDPROC                               <DQ>
3610 PROC FARBE                            <PQ>
3620 DPOKE 88,AF:POKE 87,9:COLOR COL       <HK>
3630 ENDPROC                               <OZ>
3640 PROC AN                               <HN>
3650 D=USR($0600,65):POKE 559,58           <RE>
3660 ENDPROC                               <PI>
3670 PROC AUS                              <AP>
3680 POKE 559,%0:D=USR($0600,%0):PAUSE %
3                                          <RZ>
3690 ENDPROC                               <PR>
3700 PROC WAIT_KEY                         <NW>
3710 REPEAT                                <ES>
3720 EXEC MAUS                             <LQ>
3730 UNTIL PEEK(764)<255 OR STRIG(%0)=%0   <OC>
3740 IF PEEK(764)<255:GET T:ELSE :T=%0:E
NDIF                                       <QN>
3750 ENDPROC                               <PH>
3760 PROC SAFE                             <DP>
3770 MOVE AH,UN,$1E00                      <RY>
3780 ENDPROC                               <PQ>
3790 PROC UNSAFE                           <TZ>
3800 MOVE UN,AH,$1E00                      <OF>
3810 ENDPROC                               <OX>
3820 PROC MENUE_INIT                       <VC>
3830 EXEC MENUE:POKE 752,1                 <MJ>
3840 ? "Plt▲Lne▲Box▲Zrc▲Txt▲Scl▲Fil▲Cut▲
Ray▲Kln▲Ers▲Dsk▲New▲"                      <GJ>
3851 POSITION 21,%1:? "Color▲No.:▲"        <EW>
3870 CH=%1:POKE 87,9:DPOKE 88,ME+80        <YC>
3880 FOR I=%0 TO 15                        <MW>
3890 COLOR I:TEXT I*5,%0,"▲"               <XS>
3910 NEXT I                                <FM>
3920 ENDPROC                               <PC>
4120 PROC PLOT                             <TC>
4130 WHILE STRIG(%0)=%0 AND Y>-%1          <QN>
4140 EXEC HELL:PLOT X,Y:EXEC FARBE:PLOT
X,Y:EXEC MAUS                              <PC>
4190 WEND                                  <UK>
4200 ENDPROC                               <OJ>
4210 PROC FIL                              <SS>
4220 IF COL>%0 AND LUM>%0                  <IT>
4230 EXEC HELL:PAINT X,Y:EXEC FARBE:PAIN
T X,Y                                      <DH>
4240 ENDIF                                 <SZ>
4260 ENDPROC                               <PB>
4270 PROC LNE                              <UT>
4280 WHILE STRIG(%0) AND Y>-%1 AND PEEK(
764)<>28                                   <VN>
4290 PLOT XA,YA:DRAWTO X,Y:EXEC PARTSAFE
:EXEC MAUS                                 <MA>
4310 WEND                                  <TQ>
4320 IF Y>-%1 AND PEEK(764)<>28 THEN PLO
T XA,YA:DRAWTO X,Y:EXEC FARBE:PLOT XA,YA
:DRAWTO X,Y:PAUSE %3                       <SQ>
4340 ENDPROC                               <OX>
4350 PROC BOX                              <AI>
4400 WHILE STRIG(%0) AND Y>-%1 AND PEEK(
764)<>28                                   <UT>
4420 PLOT XA,YA:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <YP>
4430 EXEC PARTSAFE:EXEC MAUS               <RS>
4440 WEND                                  <UB>
4450 IF Y>-%1 AND PEEK(764)<>28            <LL>
4470 PLOT XA,YA:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <ZE>
4472 IF D=%0                               <AO>
4480 EXEC FARBE:DRAWTO XA,Y:DRAWTO X,Y:D
RAWTO X,YA:DRAWTO XA,YA                    <SI>
4482 ENDIF                                 <TX>
4490 ENDIF                                 <TS>
4500 ENDPROC                               <OP>
4510 PROC CRC                              <RI>
4520 WHILE STRIG(%0) AND Y>%0 AND PEEK(7
64)<>28                                    <GG>
4530 CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y):EX
EC UNSAFE:EXEC MAUS                        <MI>
4550 WEND                                  <UG>
4560 IF Y>-%1 AND PEEK(764)<>28            <LQ>
4580 CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y):EX
EC FARBE:CIRCLE XA,YA,ABS(XA-X),ABS(YA-Y
)                                          <FF>
4590 ENDIF                                 <TU>
4600 ENDPROC                               <OR>
4610 PROC TXT                              <JW>
4620 L=10:G=%1:POS=%0:NAME$="":IF Y<184
THEN EXEC INPUT                            <LZ>
4630 WHILE STRIG(%0) AND Y>%0 AND Y<184
AND PEEK(764)<>28                          <GS>
4640 TEXT X,Y,NAME$:MOVE UN+Y*40,AH+Y*40
,320:EXEC MAUS                             <GX>
4660 WEND                                  <UL>
4670 IF Y<%0 OR PEEK(764)=28:EXEC UNSAFE   <QA>
4680 ELSE :TEXT X,Y,NAME$:EXEC FARBE:TEX
T X,Y,NAME$:WHILE STRIG(%0)=%0:WEND :END
IF                                         <LO>
4690 ENDPROC                               <PS>
4700 PROC SCL                              <VS>
4710 D=%1:EXEC BOX:P1=XA:P2=YA:P3=X:P4=Y   <EA>
4712 IF PEEK(764)<>28 AND X>XA AND Y>YA    <AX>
4720 EXEC UNSAFE:REPEAT :EXEC MAUS:UNTIL
 STRIG(%0)=%0:XA=X:YA=Y:EXEC HELL          <EJ>
4730 PAUSE 10:EXEC BOX:P5=XA:P6=YA:P7=X:
P8=Y                                       <WW>
4732 IF PEEK(764)<>28 AND X>XA AND Y>YA    <BD>
4740 EXEC UNSAFE:EXEC SAFE:EXEC AUS        <GK>
4750 Z1=(P1-P3)/(P5-P7):Z2=(P2-P4)/(P6-P
8)                                         <IU>
4762 FOR I=%0 TO %1                        <BR>
4770 FOR I1=P5 TO P7                       <WC>
4780 FOR I2=P6 TO P8                       <XY>
4810 DPOKE 88,UN:LOCATE P1+(I1-P5)*Z1,P2
+(I2-P6)*Z2,Z:COLOR Z:DPOKE 88,AH+I*$200
0                                          <RK>
4820 DPOKE 88,AH+I*$2000:PLOT I1,I2        <JS>
4850 NEXT I2                               <CZ>
4860 NEXT I1                               <CQ>
4870 MOVE AF,UN,7680                       <KT>
4880 NEXT I                                <GG>
4882 EXEC AN                               <EM>
4884 ELSE :EXEC UNSAFE:ENDIF               <IZ>
4886 ELSE :EXEC UNSAFE:ENDIF               <JH>
4890 ENDPROC                               <PW>
4900 PROC DSK                              <WK>
4920 EXEC MENUE:? "Save▲Load▲Frmt▲Dir▲▲E
Dir▲Bye▲▲Quit"                             <GJ>
4930 REPEAT                                <FD>
4940 EXEC WAIT_KEY:D=%0:T=T&95             <AL>
4950 IF T=%0                               <FY>
```

Listing. »Paint256« (Fortsetzung)

```
4951 IF Y<-23 THEN D=X DIV 10+%1              <YQ>
4960 ELSE                                     <UH>
4970 IF T=83 THEN D=%1                        <LY>
4980 IF T=76 THEN D=%2                        <NW>
4990 IF T=70 THEN D=%3                        <MB>
5000 IF T=68 THEN D=4                         <IP>
5010 IF T=69 THEN D=5                         <JZ>
5020 IF T=66 THEN D=6                         <JN>
5030 IF T=81 THEN D=7                         <IZ>
5040 ENDIF                                    <SW>
5060 EXEC MENUE:IF D<6 THEN ON D EXEC SA
VE,LAD,FORMAT,NDIR,EDIR:D=6                   <EK>
5070 EXEC MENUE_INIT                          <RD>
5080 UNTIL D>5                                <RE>
5090 IF D=7 THEN EXEC AUS:GRAPHICS %0:PO
KE 53251,%0:END                               <AC>
5100 ENDPROC                                  <OI>
5110 PROC RUB                                 <YH>
5120 MAUS$="{CTL ,}{ESC CTL >}U+U+U+{ESC
 CTL >}{CTL ,}{CTL ,}":POKE 87,15             <CL>
5130 WHILE Y>%0 AND PEEK(764)=255             <IU>
5140 WHILE STRIG(%0)=%0 AND Y>%1              <LS>
5150 EXEC MAUS:COLOR %0                       <UZ>
5160 DPOKE 88,AH:TEXT X*%2,Y-%1,"▲"           <BP>
5180 DPOKE 88,AF:TEXT X*%2,Y-%1,"▲"           <AP>
5200 WEND                                     <TM>
5210 EXEC MAUS                                <LF>
5220 WEND                                     <TS>
5230 MAUS$="{CTL ,}{CTL ,}@p{SHIFT =}x<.
{CTL G}{CTL B}{CTL ,}"                        <JK>
5240 ENDPROC                                  <OW>
5250 PROC NEW                                 <AG>
5270 EXEC MENUE:? "▲▲Clear▲screen▲(Yes/N
o)▲?":EXEC WAIT_KEY                           <SV>
5290 IF T=89                                  <SC>
5300 POKE AF,%0:MOVE AF,AF+%1,$1E00:MOVE
 AF,AH,$1E00                                  <FK>
5330 ENDIF                                    <SZ>
5340 EXEC MENUE_INIT                          <RA>
5350 ENDPROC                                  <PB>
5360 PROC FORMAT                              <ZV>
5370 EXEC MENUE:? "O.k.to▲format▲disk▲in
▲drive▲1?▲(Yes/No)"                           <WP>
5380 EXEC WAIT_KEY                            <KO>
5390 IF T=89                                  <SE>
5400 EXEC AUS:XIO 254,#%1,%0,%0,"D1":EXE
C AN                                          <TZ>
5430 ENDIF                                    <TB>
5440 ENDPROC                                  <PA>
5450 PROC NDIR                                <LC>
5460 LEER$="D:*.PIC":L=10:EXEC DIR:LEER$
="{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CT
L ,}{CTL ,}"                                  <MU>
5490 ENDPROC                                  <PP>
5500 PROC EDIR                                <GU>
5510 LEER$="D:*.*":L=13:EXEC DIR:LEER$="
{CTL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}"          <KJ>
5540 ENDPROC                                  <PC>
5550 PROC DIR                                 <VP>
5560 REPEAT                                   <FF>
5570 EXEC AUS:CLOSE #%1:OPEN #%1,6,%0,LE
ER$                                           <QM>
5590 REPEAT                                   <FO>
5610 EXEC AUS:INPUT #%1,NAME$:SL=11           <LI>
5612 IF NAME$(5)<>"FREE▲SECTORS"              <RL>
5620 POSITION 10,%0                           <DF>
5640 FOR I=%3 TO 10                           <KE>
5650 IF NAME$(I,I)="▲" THEN SL=SL-%1          <DR>
5660 NEXT I                                   <FX>
5670 IF NAME$(%2,%2)="▲":? NAME$(%3,L)        <EQ>
5680 ELSE :? NAME$(%1,15):ENDIF               <JQ>
5690 EXEC AN                                  <EE>
5700 WHILE STICK(%0)=15 AND PEEK(764)=25
5 AND STRIG(%0):WEND                          <WW>
5710 IF PEEK(764)<255 THEN GET T              <FP>
5712 ENDIF                                    <TJ>
5720 UNTIL T=155 OR STRIG(%0)=%0 OR NAME
$(%2,%2)<>"▲" OR T=27                         <GW>
5730 UNTIL STRIG(%0)=%0 OR T=27 OR T=155      <UD>
5740 CLOSE #%1                                <HX>
5750 ENDPROC                                  <PJ>
```

```
5760 PROC MAUS                                <QE>
5770 A=STICK(%0)                              <KC>
5780 X=X+XX*(A&8=%0)-(A&4=%0)                 <QO>
5790 Y=Y+(A&%2=%0)-(A&%1=%0)                  <RD>
5800 IF Y>191 THEN Y=191                      <IN>
5810 IF Y<-32 THEN Y=-32                      <WD>
5820 IF X>79 THEN X=79                        <ZJ>
5830 IF X<%0 THEN X=%0                        <QU>
5850 POKE 53251,X*%2+48:MOVE ADR(MAUS$),
PMB+Y,11                                      <TC>
5860 ENDPROC                                  <PO>
5870 # FEHLER                                 <NB>
5890 CLOSE #%1:EXEC MENUE:EXEC AN             <XP>
5900 ? "▲▲▲Fehler▲Nr.:▲";ERR;".▲▲Taste/K
nopf▲"                                        <UC>
5910 EXEC WAIT_KEY:EXEC MENUE_INIT            <PQ>
5930 GO# START                                <JB>
5940 PROC INPUT                               <PL>
5950 IF NOT (G) THEN EXEC MENUE:POSITION
 2,0:? "Save▲as?":NAME$(%1)=""                <FC>
5960 REPEAT                                   <FN>
5970 IF G:MOVE UN+Y*40,AH+Y*40,320:TEXT
X,Y,NAME$                                     <DL>
5980 ELSE :POSITION 11,0:? NAME$;"▲":END
IF                                            <ZW>
5990 GET T:D=T&127                            <VG>
6010 IF (D>-%1 AND D<27) OR (D>31 AND D<
125)                                          <ZN>
6020 POS=POS+(POS<L):NAME$(POS,POS)=CHR$
(T)                                           <UY>
6040 ENDIF                                    <SX>
6050 IF T=126 AND POS=%0 THEN NAME$=""        <UI>
6060 IF T=126 AND POS>%0 THEN POS=POS-%1
:NAME$(POS+%1)="^"                            <TZ>
6070 UNTIL T=155 OR T=27                      <RQ>
6072 IF T=27 THEN POKE 764,28                 <KE>
6080 ENDPROC                                  <PF>
6090 PROC LAD                                 <OK>
6100 EXEC NDIR                                <GI>
6110 IF T<>27 OR NAME$(%2,%2)<>"▲"            <GJ>
6120 NAME$(%1,%2)="D:":NAME$(SL)=".PIC"       <QH>
6130 EXEC AUS:OPEN #%1,4,%0,NAME$             <CO>
6140 BGET #%1,AH,$1E00:BGET #%1,AF,$1E00
:CLOSE #%1:EXEC AN                            <OC>
6170 ENDIF                                    <TI>
6190 ENDPROC                                  <PK>
6200 PROC SAVE                                <KQ>
6210 G=%0:L=8:POS=%0:EXEC INPUT               <ZM>
6220 IF T<>27                                 <RT>
6230 -MOVE ADR(NAME$(%1)),ADR(NAME$(%3))
,8                                            <OU>
6240 NAME$(%1,%2)="D:":NAME$(POS+%3)=".P
IC"                                           <TO>
6250 EXEC AUS:OPEN #%1,8,%0,NAME$             <GO>
6260 BPUT #%1,AH,$1E00:BPUT #%1,AF,$1E00
:CLOSE #%1:EXEC AN                            <UK>
6290 ENDIF                                    <TQ>
6300 ENDPROC                                  <ON>
7000 PROC CUT                                 <AV>
7010 D=%1:EXEC BOX:EXEC UNSAFE:P2=Y:XA=X
A DIV %2:P1=ABS(X DIV %2-XA):EXEC SAFE:X
X=%2                                          <YJ>
7020 IF Y>YA AND X>XA                         <GE>
7100 REPEAT                                   <EH>
7110 EXEC UNSAFE                              <OT>
7120 FOR I=YA TO P2                           <LO>
7122 IF (Y+I)*40-P2*40<%0 THEN POKE 764,
28:EXIT                                       <EM>
7130 MOVE XA+UN+I*40,X DIV %2+AH+(Y+I)*4
0-P2*40-P1+%1,P1                              <UV>
7140 NEXT I                                   <FJ>
7150 EXEC MAUS                                <LR>
7160 UNTIL STRIG(%0)=%0 OR PEEK(764)=28       <JB>
7170 IF PEEK(764)<>28                         <FM>
7180 MOVE AF,UN,7680                          <KL>
7190 FOR I=YA TO P2                           <MJ>
7200 MOVE XA+UN+I*40,X DIV %2+AF+(Y+I)*4
0-P2*40-P1+%1,P1                              <SC>
7210 NEXT I                                   <FC>
```

Listing. »Paint256« (Fortsetzung)

```
7220 ELSE                                   <TK>
7230 EXEC UNSAFE                            <PB>
7240 ENDIF                                  <TC>
7250 ENDIF                                  <TF>
7260 XX=%1                                  <QE>
7290 ENDPROC                                <PN>
7300 PROC RAY                               <AR>
7310 REPEAT                                 <EO>
7320 EXEC LNE:EXEC HELL:EXEC SAFE:IF CH=
10 THEN XA=X:YA=Y                           <WB>
7350 UNTIL PEEK(764)=28 OR Y<%1             <NH>
7390 ENDPROC                                <PP>
```

Listing. »Paint256« (Schluß)

```
1000 REM Mandelbrotmenge/Apfelmaennchen     <BD>
1010 REM in 256 Farben.                     <KC>
1020 --                                     <VK>
1030 REM Voraussetzung:                     <RJ>
1040 REM Atari XL/XE                        <EA>
1050 REM Floppy (Turbo Basic XL)            <LF>
1060 REM Grafikerweiterung fuer 256 Farb
en                                          <EY>
1070 REM (COL256.BIN auf Diskette in Lau
fwerk 1)                                    <WX>
1080 --                                     <WC>
1090 REM (c) by Stephan Gerle               <GN>
1100 --                                     <VG>
1110 EXEC FARB256_INIT                      <NH>
1120 GRAPHICS %0                            <YG>
1130 DIM LU$(12),RO$(12),FILE$(20),LEER$
(256)                                       <UG>
1140 LU$="{CTL ,}{CTL ,}{CTL ,}{CTL ,}{C
TL ,}{CTL ,}{CTL ,}{CTL ,}{CTL ,}{ESC CT
L >}{CTL ,}{CTL ,}":RO$="{CTL ,}{CTL ,}{
ESC CTL >}{CTL A}{CTL A}{CTL A}{CTL A}{C
TL A}{CTL A}{CTL A}{CTL ,}{CTL ,}"          <HJ>
1150 # MENUE_NEU                            <FA>
1160 POKE 82,%2:CLS                         <RJ>
1170 SETCOLOR %1,%0,12:SETCOLOR %2,%0,%0    <PA>
1180 ? "▲▲Mandelbrotmenge▲(Apfelmaennche
n)"                                         <VL>
1190 ? "nach▲Spektrum▲der▲Wissenschaft▲1
0/86"                                       <AP>
1200 ? "▲▲▲▲▲▲▲▲(c)▲by▲Stephan▲Gerle"       <BQ>
1210 POKE 82,5:?                            <CK>
1220 ? "Laden▲Bild"                         <XD>
1230 ? "Speichern▲Bild"                     <PA>
1240 ? "Berechnen▲Bild"                     <FW>
1250 ? "Ausschnitt▲vom▲Bild"                <SZ>
1260 ? "Directory"                          <NA>
1270 ?                                      <WU>
1280 ? "Umschalten▲zum▲Bild:▲SELECT"        <MG>
1290 ? "Vom▲Bild▲zum▲Menue▲:▲START"         <FA>
1300 ?                                      <WB>
1310 ? "Aktuelle▲Parameter:"                <QI>
1320 ? "-------------------"                <LH>
1330 ? "Real-Teil▲▲▲▲▲▲(x)▲:▲";AECKE        <HO>
1340 ? "Imaginaer-Teil▲(y)▲:▲";BECKE        <OA>
1350 ? "Seitenlaenge▲▲▲(s)▲:▲";SEITE        <OA>
1360 ?                                      <WT>
1370 ? "Ihre▲Wahl?▲";                       <PR>
1380 MENUE=DPEEK(88)                        <FV>
1390 MENUEDLST=DPEEK(560)                   <GS>
1400 # START                                <MT>
1410 DPOKE 560,MENUEDLST                    <HG>
1420 DPOKE 88,MENUE:POKE 87,%0              <FN>
1430 POSITION 22,19:? "";                   <AM>
1440 REPEAT                                 <ET>
1450 UNTIL PEEK($D01F)=5 OR PEEK(764)<25
5                                           <CJ>
1460 IF PEEK($D01F)=5                       <DP>
1470 D=USR($0600,64)                        <HW>
1480 WHILE PEEK($D01F)<>6:WEND              <JU>
1490 D=USR($0600,%0)                        <TK>
1500 ELSE                                   <TE>
1510 GET TT:TT=TT&95:T=%0                   <BQ>
1520 IF CHR$(TT)="S" THEN EXEC SPEICHERN    <DP>
1530 IF CHR$(TT)="L" THEN EXEC LADEN        <BE>
1540 IF CHR$(TT)="B" THEN EXEC EINGABE:E
XEC RECHNEN:T=%1                            <EW>
1550 IF CHR$(TT)="A" THEN EXEC AUSSCHNIT
T:EXEC RECHNEN:T=%1                         <XH>
1560 IF CHR$(TT)="D" THEN ? :DIR :GET TT
:T=%1                                       <AO>
1570 IF T=%1 THEN POP :GO# MENUE_NEU        <FW>
1580 ENDIF                                  <TO>
1590 GO# START                              <JH>
1600 --                                     <VQ>
1610 PROC EINGABE                           <YY>
1620 ?                                      <WN>
1630 ? "Realteil▲der▲linken▲unteren▲Ecke
"                                           <KC>
1640 INPUT "X=",AECKE                       <JF>
1650 ? "Imaginaerteil▲der▲linken▲unteren
▲Ecke"                                      <YG>
1660 INPUT "Y=",BECKE                       <KP>
1670 ? "Seitenlaenge▲des▲Ausschnitts"       <BR>
1680 INPUT "S=",SEITE                       <EG>
1690 ? "Maximale▲Iterationstiefe"           <RP>
1700 INPUT "T=",TIEFE                       <TQ>
1710 ENDPROC                                <OT>
1720 --                                     <VY>
1730 PROC RECHNEN                           <NE>
1740 D=USR($0600,64)                        <HT>
1750 POKE AH,%0:MOVE AH,AH+%1,$1DFF         <SR>
1760 MOVE AH,AF,$1E00                       <CJ>
1770 SPALT=SEITE/160                        <VF>
1780 POKE 87,9:DPOKE 88,AH:COLOR 8          <CD>
1790 TEXT %0,%0,STR$(AECKE)                 <DC>
1800 TEXT %0,8,STR$(BECKE)                  <IL>
1810 TEXT %0,16,STR$(SEITE)                 <PP>
1820 DPOKE 88,AF:COLOR 6                    <VB>
1830 TEXT %0,%0,STR$(AECKE)                 <CM>
1840 TEXT %0,8,STR$(BECKE)                  <IX>
1850 TEXT %0,16,STR$(SEITE)                 <QB>
1860 FOR M=%0 TO 159                        <FY>
1870 BC=M*SPALT+BECKE                       <WQ>
1880 FOR N=%0 TO 159 STEP %2                <ZV>
1890 AC=N*SPALT+AECKE                       <WI>
1900 AZ=%0:BZ=%0:ZAEHLER=%0                 <TW>
1910 REPEAT                                 <EU>
1920 AQ=AZ:A2=AZ^%2:B2=BZ^%2                <HC>
1930 AZ=A2-B2+AC                            <PV>
1940 BZ=%2*AQ*BZ+BC                         <BF>
1950 ZAEHLER=ZAEHLER+%1                     <MD>
1960 UNTIL (A2+B2>4) OR (ZAEHLER>TIEFE)     <LO>
1970 IF ZAEHLER<=TIEFE                      <TB>
1980 COLOR ZAEHLER MOD 16                   <JQ>
1990 DPOKE 88,AH                            <BI>
2000 PLOT N DIV %2,191-M                    <WT>
2010 COLOR (ZAEHLER DIV 16) MOD 16          <AS>
2020 DPOKE 88,AF                            <YQ>
2030 PLOT N DIV %2,191-M                    <XC>
2040 ENDIF                                  <ST>
2050 IF PEEK($D01F)<>7                      <EG>
2060 IF PEEK($D01F)=6                       <ED>
2070 POKE 559,$22:D=USR($0600,64)           <YB>
2080 ELSE                                   <TT>
2090 IF PEEK($D01F)=5                       <DR>
2100 POKE 559,%0:D=USR($0600,%0)            <IX>
2110 ENDIF                                  <SM>
2120 ENDIF                                  <SP>
2130 ENDIF                                  <SS>
2140 NEXT N                                 <HH>
2150 NEXT M                                 <GZ>
2160 D=USR($0600,%0):GRAPHICS %0            <HX>
2170 ENDPROC                                <PA>
2180 --                                     <HF>
2190 PROC SPEICHERN                         <TW>
2200 ?                                      <WA>
2210 ? "Nummer▲des▲Bildes▲";                <QA>
2220 INPUT NR                               <FV>
2230 FILE$="D:APFEL"                        <WV>
2240 FILE$(LEN(FILE$)+%1)=STR$(NR)          <CH>
2250 FILE$(LEN(FILE$)+%1)=".PIC"            <FK>
2260 TRAP #FEHLER                           <XI>
2270 OPEN #%1,8,%0,FILE$                    <CK>
```

Listing. »Apfel256«

```
2280 BPUT #%1,AH,$1E00                   <TY>
2290 BPUT #%1,AF,$1E00                   <SV>
2300 ? #%1,STR$(AECKE)                   <UI>
2310 ? #%1,STR$(BECKE)                   <VC>
2320 ? #%1,STR$(SEITE)                   <RQ>
2330 # FEHLER                            <MC>
2340 CLOSE #%1                           <HM>
2350 POP :GO# MENUE_NEU                  <NH>
2360 ENDPROC                             <PB>
2370 --                                  <WG>
2380 PROC LADEN                          <SN>
2390 ?                                   <XD>
2400 ? "Nummer▲des▲Bildes▲";             <QB>
2410 INPUT NR                            <FW>
2420 FILE$="D:APFEL"                     <WW>
2430 FILE$(LEN(FILE$)+%1)=STR$(NR)       <CI>
2440 FILE$(LEN(FILE$)+%1)=".PIC"         <FL>
2450 TRAP #NICHT_VORHANDEN               <NK>
2460 POKE 73,%0                          <HD>
2470 OPEN #%1,4,%0,FILE$                 <AG>
2480 BGET #%1,AH,$1E00                   <MT>
2490 BGET #%1,AF,$1E00                   <LQ>
2500 INPUT #%1,FILE$                     <CU>
2510 AECKE=VAL(FILE$)                    <CO>
2520 INPUT #%1,FILE$                     <DA>
2530 BECKE=VAL(FILE$)                    <DA>
2540 INPUT #%1,FILE$                     <DG>
2550 SEITE=VAL(FILE$)                    <MD>
2560 # NICHT_VORHANDEN                   <XS>
2570 CLOSE #%1                           <HZ>
2580 POP :GO# MENUE_NEU                  <NU>
2590 ENDPROC                             <PO>
2600 --                                  <VR>
2610 PROC AUSSCHNITT                     <RB>
2620 D=USR($0600,65)                     <IF>
2630 PMB=112:POKE 54279,PMB              <VA>
2640 PMB=PMB*256+1024+214                <PF>
2650 POKE 559,58                         <CW>
2660 POKE 53256,%0:POKE 53257,%0         <WB>
2670 POKE 704,255:POKE 705,255           <EP>
2680 POKE 53277,%2                       <FH>
2690 X=%0:Y=%0                           <PK>
2700 AB=8                                <SE>
2710 MOVE ADR(LEER$),PMB-214,256         <TJ>
2720 MOVE ADR(LEER$),PMB+256-214,256     <ZC>
2730 POKE 87,9:DPOKE 88,AH:COLOR 8       <BP>
2740 AA=AECKE:BB=BECKE:SPALT=SEITE/80    <BA>
2750 REPEAT                              <FD>
2760 POKE AH,%0                          <VH>
2770 MOVE AH,AH+%1,960                   <LV>
2780 TEXT %0,%0,STR$(AECKE)              <DA>
2790 TEXT %0,8,STR$(BECKE)               <JL>
2800 TEXT %0,16,STR$(SEITE)              <PN>
2810 A=STICK(%0)                         <JJ>
2820 IF STRIG(%0)                        <XG>
2830 X=X+(A&8=%0)-(A&4=%0)               <WR>
2840 Y=Y-(A&%2=%0)+(A&%1=%0)             <PT>
2850 IF X<%0 THEN X=%0                   <QX>
2860 IF X+AB>80 THEN X=X-%1              <IX>
2870 IF Y<%0 THEN Y=%0                   <SF>
2880 IF Y+AB>80 THEN Y=Y-%1              <LG>
2890 ELSE                                <UM>
2900 AB=AB+(A&8=%0)-(A&4=%0)             <KW>
2910 IF AB<%1 THEN AB=%1                 <BH>
2920 IF X+AB>80 THEN AB=AB-%1            <DD>
2930 IF Y+AB>80 THEN AB=AB-%1            <DP>
2940 ENDIF                               <TL>
2950 POKE 53248,X*%2+48                  <RS>
2960 POKE 53249,X*%2+40+AB*2             <QZ>
2970 MOVE ADR(LU$),PMB-Y*%2,12           <GD>
2980 MOVE ADR(RO$),PMB-Y*%2-%2*AB+264,12 <SP>
2990 AECKE=AA+X*SPALT                    <GR>
3000 BECKE=BB+Y*SPALT                    <GT>
3010 SEITE=AB*SPALT                      <LS>
3020 UNTIL PEEK($D01F)=6                 <YJ>
3030 POKE 53248,%0:POKE 53249,%0         <XB>
3040 D=USR($0600,%0):GRAPHICS %0         <HQ>
3050 ? "Maximale▲Iterationstiefe"        <QT>
3060 INPUT TIEFE                         <UL>
3070 ENDPROC                             <OZ>
3080 --                                  <WE>
3090 PROC FARB256_INIT                   <SF>
3100 REM Grafik mit 256 Farben           <UB>
3110 REM Aufruf mit D=USR($600,GTIAMode) <TB>
3120 REM Ausschalten mit D=USR($600,0)   <YS>
3130 REM Helligkeitsbild bei AH=$8010    <FJ>
3140 REM Farbbild        bei AF=$A010    <CR>
3150 AH=$8010:AF=$A010                   <AS>
3160 POKE AH,%0:MOVE AH,AH+%1,$1DFF      <SK>
3170 MOVE AH,AF,$1E00                    <CC>
3180 POKE 106,120:REM Speicher schuetzen <JX>
3190 --                                  <WJ>
3200 REM Displaylist erstellen           <MW>
3210 --                                  <VN>
3220 A=$7800:ADA=AH:ADB=AF+40            <ZL>
3230 POKE A,112:POKE A+%1,240:POKE A+%2,
240                                      <KF>
3240 FOR I=A+%3 TO A+%3+(191*%3) STEP 6  <UZ>
3250 POKE I,$CF                          <KD>
3260 DPOKE I+%1,ADA                      <ES>
3270 ADA=ADA+80                          <DG>
3280 POKE I+%3,$CF                       <XI>
3290 DPOKE I+4,ADB                       <CG>
3300 ADB=ADB+80                          <DH>
3310 NEXT I                              <FA>
3320 POKE I,65                           <WY>
3330 DPOKE I+%1,A                        <NW>
3340 A=$7C00:ADA=AF:ADB=AH+40            <EN>
3350 POKE A,112:POKE A+%1,112:POKE A+%2,
240                                      <IM>
3360 FOR I=A+%3 TO A+%3+(191*%3) STEP 6  <VH>
3370 POKE I,$CF                          <KL>
3380 DPOKE I+%1,ADA                      <FA>
3390 ADA=ADA+80                          <DO>
3400 POKE I+%3,$CF                       <WO>
3410 DPOKE I+4,ADB                       <BM>
3420 ADB=ADB+80                          <DP>
3430 NEXT I                              <FI>
3440 POKE I,65                           <XG>
3450 DPOKE I+%1,A                        <OE>
3460 BLOAD "D:COL256.BIN":REM Grafikerwe
iterung laden                            <EF>
3470 ENDPROC                             <PH>
3480 --                                  <WM>
```

Listing. »Apfel256« (Schluß)

```
10000 PROC FARB256_INIT                  <VB>
10010 REM Grafik mit 256 Farben          <EO>
10020 REM Aufruf mit D=USR($600,GTIAMode
)                                        <BF>
10030 REM Ausschalten mit D=USR($600,0)  <CX>
10040 REM Helligkeitsbild bei AH=$8010   <MN>
10050 REM Farbbild        bei AF=$A010   <PL>
10060 AH=$8010:AF=$A010                  <VI>
10070 POKE AH,%0:MOVE AH,AH+%1,$1DFF     <TU>
10080 MOVE AH,AF,$1E00                   <XM>
10090 POKE 106,120:REM Speicher schuetze
n                                        <ZO>
10100 --                                 <JF>
10110 REM Display-List aufbauen          <VC>
10120 --                                 <JN>
10130 A=$7800:ADA=AH:ADB=AF+40           <MP>
10140 POKE A,112:POKE A+%1,240:POKE A+%2
,240                                     <ZS>
10150 FOR I=A+%3 TO A+%3+(191*%3) STEP 6 <ZD>
10160 POKE I,$CF                         <SW>
10170 DPOKE I+%1,ADA                     <WA>
10180 ADA=ADA+80                         <LG>
10190 POKE I+%3,$CF                      <LF>
10200 DPOKE I+4,ADB                      <QZ>
10210 ADB=ADB+80                         <LD>
10220 NEXT I                             <FX>
10230 POKE I,65                          <DB>
10240 DPOKE I+%1,A                       <ZZ>
```

Listing. »FARB 256«



```
10250 A=$7C00:ADA=AF:ADB=AH+40                 <SF>
10260 POKE A,112:POKE A+%1,112:POKE A+%2
,240                                           <YA>
10270 FOR I=A+%3 TO A+%3+(191*%3) STEP 6       <ZO>
10280 POKE I,$CF                               <TH>
10290 DPOKE I+%1,ADA                           <WL>
10300 ADA=ADA+80                               <KG>
10310 POKE I+%3,$CF                            <KF>
10320 DPOKE I+4,ADB                            <RK>
10330 ADB=ADB+80                               <LO>
10340 NEXT I                                   <GI>
10350 POKE I,65                                <DM>
10360 DPOKE I+%1,A:GRAPHICS 0                  <RG>
10370 BLOAD "D:COL256.BIN":REM Grafikerw
eiterung laden                                 <IL>
10380 ENDPROC                                  <VM>
```

Listing. »FARB 256« (Schluß)

```
100 EXEC FARB256_INIT                          <LO>
110 D=USR($0600,64)                            <RT>
120 POKE 106,120                               <NE>
130 COL=3:LUM=6                                <OM>
140 DPOKE 88,AH                                <QN>
150 POKE 87,9                                  <OE>
160 FOR I=0 TO 79                              <AB>
170 COLOR INT(I/5)                             <HR>
180 PLOT I,0:DRAWTO I,191                      <MF>
190 NEXT I                                     <GJ>
200 DPOKE 88,AF                                <PC>
210 FOR I=0 TO 191                             <DY>
220 COLOR INT(I/12)                            <OX>
230 PLOT 0,I:DRAWTO 79,I                       <ZM>
240 NEXT I                                     <GA>
250 OPEN #1,4,0,"K:"                           <PL>
260 GET #1,J:CLOSE #1                          <RU>
270 D=USR($0600,0)                             <CT>
280 GRAPHICS 0:END                             <IH>
290 --                                         <JV>
300 PROC FARB256_INIT                          <PB>
310 AH=$8010:AF=$A010                          <HO>
320 POKE AH,%0:MOVE AH,AH+%1,$1DFF             <SM>
330 MOVE AH,AF,$1E00                           <IE>
340 POKE 106,120                               <NK>
350 A=$7800:ADA=AH:ADB=AF+40                   <OG>
360 POKE A,112:POKE A+%1,240:POKE A+%2,2
40                                             <WR>
370 FOR I=A+%3 TO A+%3+(191*%3) STEP 6         <SU>
380 POKE I,$CF                                 <DJ>
390 DPOKE I+%1,ADA                             <PJ>
400 ADA=ADA+80                                 <WM>
410 POKE I+%3,$CF                              <KR>
420 DPOKE I+4,ADB                              <NI>
430 ADB=ADB+80                                 <XK>
440 NEXT I                                     <GC>
450 POKE I,65                                  <SU>
460 DPOKE I+%1,A                               <DS>
470 A=$7C00:ADA=AF:ADB=AH+40                   <SU>
480 POKE A,112:POKE A+%1,112:POKE A+%2,2
40                                             <UX>
490 FOR I=A+%3 TO A+%3+(191*%3) STEP 6         <SZ>
500 POKE I,$CF                                 <CV>
510 DPOKE I+%1,ADA                             <OV>
520 ADA=ADA+80                                 <WR>
530 POKE I+%3,$CF                              <KW>
540 DPOKE I+4,ADB                              <NN>
550 ADB=ADB+80                                 <XP>
560 NEXT I                                     <GH>
570 POKE I,65                                  <SZ>
580 DPOKE I+%1,A:GRAPHICS 0                    <QN>
590 BLOAD "D:COL256.BIN"                       <LD>
600 ENDPROC                                    <KI>
```

Demonstrationsprogramm für 256 Farben

Zeichensätze selbstgestrickt

Benötigen Sie für Ihr Programm ein Copyright-Zeichen? Kein Problem. Mit unserem Programm lassen sich Zeichen einfach selbst entwerfen und in eigene Basic-Programme einbinden.

Stellen Sie sich folgende Szene vor: Sie sitzen an Ihrem Computer und tippen gerade die letzten Zeilen Ihres neuen Programms ein. Da durchfährt es Sie wie ein Blitz: Sämtliche Umlaute in den begleitendenden Bildschirmtexten haben Sie als »ae«, »oe« und »ue« eingegeben. Sie taten das ganz automatisch, denn standardmäßig hat der Atari Computer keine Umlaute auf der Tastatur, und im Laufe der Zeit schreibt man halt die Umlaute nach Kreuzworträtsel-Schema. Nur, schön sieht das natürlich nicht aus.

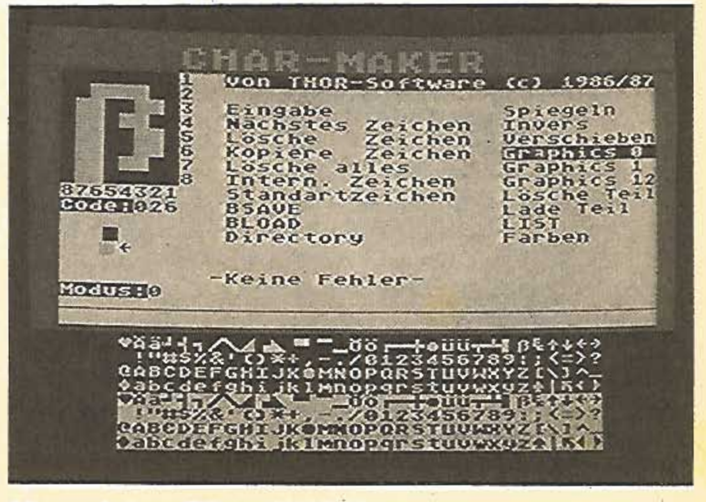
Der Atari hat zwar einen zweiten Zeichensatz eingebaut, und dieser enthält auch jede Menge verschiedener Sonderzeichen. Das deutsche »ß« ist jedoch nicht vorhanden.

Konstruieren lassen sich die Zeichen mit einem Zeichensatzprogramm. Und genau so etwas wollen wir Ihnen hier vorstellen: Ein Programm, mit dem Sie völlig problemlos eigene Zeichen malen und Zeichensätze in eigene Programme einbinden können.

Das Programm besteht aus zwei Teilen. Der erste Teil, das Hauptprogramm, ist in Turbo-Basic geschrieben. Dieses Programm benötigt ein weiteres in Maschinensprache geschriebenes Programm, das wir als AMPEL-Listing gedruckt haben. Speichern Sie es bitte auf der gleichen Diskette wie das Hauptprogramm.

Wenn Sie den »Char-Maker« starten, sehen Sie nach kurzer Zeit das Menü. In der linken oberen Ecke befindet sich das Fenster zum Malen der einzelnen Zeichen. Gleich darunter steht der Code des gerade bearbeiteten und im Fenster sichtbaren Zeichens. Unter diesem erscheint eine Anzeige, die Auskunft über die gerade eingeschaltete Farbe gibt, sowie eine Zeile, die über die verwendete Grafikstufe informiert.

Gleichzeitig leuchtet nach dem Starten des Programms in der oberen linken Ecke des Bildschirms ein Quadrat auf.



Alles auf einen Blick: »CHAR-MAKER«

```
Programmname :CHAR.OBJ
Länge        :2312 Bytes
--------------------------------

0000:FF FF 00 80 00 89 4C 49 <B9>
0008:83 4C 55 86 4C C7 86 4C <9C>
0010:4A 87 4C 73 87 4C DC 87 <17>
0018:4C 32 88 4C FA 83 4C 8C <52>
0020:88 4C A3 88 4C D1 88 4C <44>
0028:D1 88 4C D1 88 4C D1 88 <E8>
0030:4C D1 88 4C D1 88 4C D1 <66>
0038:88 00 00 00 00 00 00 00 <7C>
0040:00 00 00 00 00 00 00 00 <40>
0048:00 00 00 00 00 00 00 00 <48>
0050:00 80 01 80 02 00 03 80 <2E>
0058:03 F0 90 90 F0 90 A0 F0 <B3>
0060:47 80 80 42 00 00 02 02 <DD>
0068:02 02 02 02 02 02 02 02 <68>
0070:02 02 02 02 02 02 02 A0 <0E>
0078:20 42 A1 80 82 02 02 02 <BE>
0080:82 02 02 41 59 80 00 00 <5B>
0088:00 00 23 28 21 32 0D 2D <80>
0090:21 2B 25 32 00 00 00 00 <B3>
0098:00 00 00 00 00 00 0A 00 <AC>
00A0:34 00 0A 00 00 00 00 00 <FB>
00A8:00 00 00 00 00 00 00 00 <A8>
00B0:00 00 00 00 00 00 00 00 <B0>
00B8:00 00 00 00 00 00 00 00 <B8>
00C0:00 00 00 00 00 00 00 00 <C0>
00C8:00 00 00 00 00 00 00 00 <C8>
00D0:00 00 00 00 00 00 00 00 <D0>
00D8:00 00 00 00 00 00 00 00 <D8>
00E0:00 00 00 00 00 00 00 00 <E0>
00E8:00 00 00 00 00 00 00 00 <E8>
00F0:00 00 00 00 00 00 00 00 <F0>
00F8:00 00 00 00 00 00 00 00 <F8>
0100:00 00 00 00 00 00 00 00 <02>
0108:00 00 00 00 00 00 00 00 <0A>
0110:00 00 00 00 00 00 00 00 <12>
0118:00 00 00 00 00 00 00 00 <1A>
0120:00 00 00 00 00 00 00 00 <22>
0128:00 00 00 00 00 00 00 00 <2A>
0130:00 00 00 00 00 00 00 00 <32>
0138:00 00 00 00 00 00 00 00 <3A>
0140:00 00 00 00 00 00 00 00 <42>
0148:00 00 00 00 00 00 00 00 <4A>
0150:00 00 00 00 00 00 00 00 <52>
0158:00 00 00 00 00 00 00 00 <5A>
0160:00 00 00 00 00 00 00 00 <62>
0168:00 00 00 00 00 00 00 00 <6A>
0170:00 00 00 00 00 00 00 00 <72>
0178:00 00 00 00 00 00 00 00 <7A>
0180:00 00 00 00 00 00 00 00 <82>
0188:00 00 00 00 00 00 00 00 <8A>
0190:00 00 00 00 00 00 00 00 <92>
0198:00 00 00 00 00 00 00 00 <9A>
01A0:00 00 00 00 00 00 00 8A <2C>
01A8:0A A8 B9 4B 80 85 CB B9 <C6>
01B0:4C 80 18 6D 57 80 85 CC <54>
01B8:60 AE 78 02 A0 00 8A 29 <A0>
01C0:01 D0 01 88 8A 29 02 D0 <DD>
01C8:01 C8 8C 3A 80 A0 00 8A <8F>
01D0:29 04 D0 01 88 8A 29 08 <37>
01D8:D0 01 C8 98 AA AC 3A 80 <84>
01E0:60 AD 33 80 0A 0A 18 69 <5D>
01E8:30 AA AD 34 80 0A 0A 18 <56>
01F0:69 10 A8 60 8E 3C 80 8C <0E>
01F8:3D 80 20 DB 81 EC 3C 80 <9F>
0200:D0 05 A9 00 8D 49 80 CC <1D>
0208:3D 80 D0 0C A9 00 8D 4A <EF>
0210:80 EC 3C 80 D0 02 18 60 <F5>
0218:38 60 AD 47 80 18 6D 49 <A2>
0220:80 8D 47 80 AD 48 80 18 <1E>
0228:6D 4A 80 AE 37 80 F0 04 <DF>
0230:18 6D 4A 80 8D 48 80 A2 <F5>
0238:00 A0 02 86 CB 98 18 6D <1E>
0240:57 80 85 CC A0 7F A2 03 <6C>
0248:A9 00 91 CB 88 C0 FF D0 <98>
0250:F9 AC 48 80 BD 53 80 91 <AE>
0258:CB C8 CA 10 F7 AE 47 80 <AB>
0260:8E 00 D0 60 4C 3C 83 A5 <2A>
0268:14 48 29 10 D0 04 68 49 <16>
0270:FF 48 68 29 8F 8D 12 D0 <00>
0278:AD 35 80 D0 E7 AD 78 02 <1C>
0280:C9 0F F0 0A CD A0 80 D0 <81>
0288:05 EE 95 80 D0 0B A9 00 <31>
0290:8D 95 80 AD 78 02 8D A0 <F0>
0298:80 AD 95 80 C9 40 90 07 <20>
02A0:A9 40 8D 95 80 D0 05 A5 <32>
02A8:14 4A 90 B8 AD 38 80 F0 <C9>
02B0:1A AD 47 80 29 04 F0 13 <6A>
02B8:AD 34 80 C9 08 B0 0C AD <94>
02C0:33 80 C9 09 B0 05 A9 0B <1E>
02C8:8D 78 02 AE 47 80 AC 48 <35>
02D0:80 20 EE 81 B0 69 20 B3 <7C>
02D8:81 8E 3C 80 8C 3D 80 A9 <01>
02E0:00 8D 94 80 AD 38 80 F0 <CD>
02E8:11 AD 33 80 C9 08 B0 0A <DE>
02F0:AD 34 80 C9 08 B0 03 EE <CB>
02F8:94 80 18 AD 33 80 6D 3C <63>
0300:80 AC 94 80 F0 05 6D 3C <94>
0308:80 29 FE C9 28 90 08 A9 <1D>
0310:00 8D 3C 80 AD 33 80 8D <AF>
0318:33 80 8C 08 D0 AD 34 80 <A7>
0320:18 6D 3D 80 C9 12 90 08 <B2>
0328:A9 00 8D 3D 80 AD 34 80 <AB>
0330:8D 34 80 AD 3D 80 8D 4A <F1>
0338:80 AD 3C 80 8D 49 80 20 <EB>
0340:14 82 A9 79 8D 00 02 A9 <AF>
0348:84 8D 01 02 4C 62 E4 68 <CF>
0350:EE 35 80 A9 40 8D 0E D4 <39>
0358:AD 57 80 8D 07 D4 A9 03 <00>
0360:8D 1D D0 AD 2F 02 09 0C <58>
0368:8D 2F 02 A9 01 8D 49 80 <67>
0370:8D 4A 80 A9 01 8D 6F 02 <C6>
0378:A9 FF 8D 0C D0 A9 07 A2 <B6>
0380:82 A0 61 20 5C E4 A9 01 <DC>
0388:8D 49 80 8D 4A 80 A5 58 <72>
0390:8D 5E 80 A5 59 8D 5F 80 <72>
0398:A9 59 8D 30 02 A9 80 8D <7E>
03A0:31 02 A9 79 8D 00 02 A9 <7D>
03A8:84 8D 01 02 A9 C0 8D 0E <C1>
03B0:D4 A2 00 EE 35 80 8A 8E <2C>
03B8:94 80 20 26 84 AE 94 80 <89>
03C0:9D A1 80 E8 D0 F0 20 D5 <3F>
03C8:84 A9 00 8D 35 80 60 18 <46>
03D0:A5 58 65 CD 8D 42 80 A5 <B4>
03D8:59 69 00 8D 43 80 A5 CE <6B>
03E0:F0 13 AD 42 80 18 69 28 <D2>
03E8:8D 42 80 90 03 EE 43 80 <74>
03F0:C6 CE 18 90 E9 AD 42 80 <5B>
03F8:85 CD AD 43 80 85 CE 60 <88>
0400:68 EE 35 80 68 68 85 CD <09>
0408:68 68 85 CE 68 68 8D 3A <01>
0410:80 20 C9 83 A0 00 AD 3A <5A>
0418:80 F0 0D B1 CD 49 80 91 <2A>
0420:CD C8 CE 3A 80 18 90 EE <69>
0428:CE 35 80 60 A2 00 8E 46 <F0>
0430:80 C9 00 08 29 7F C9 20 <24>
0438:B0 04 69 40 90 07 C9 60 <BC>
0440:B0 03 38 E9 20 28 10 05 <39>
0448:09 80 EE 46 80 48 29 7F <01>
0450:A2 00 86 CC 0A 26 CC 0A <41>
0458:26 CC 0A 26 CC 85 CB A5 <B9>
0460:CC 6D 58 80 85 CC 68 60 <40>
0468:8E 0A D4 CA D0 FA 60 68 <67>
0470:D8 18 69 01 8D 00 02 68 <78>
0478:69 00 8D 01 02 68 40 48 <4D>
0480:8A 48 A2 0F 20 62 84 A2 <38>
0488:04 BD 96 80 9D 16 D0 CA <43>
```

Das ist der Cursor des Programms. Mit ihm werden die einzelnen Menüpunkte ausgewählt. Wenn Sie jetzt einen Joystick an Port 1 anschließen, können Sie den Cursor mit dem Joystick über den gesamten Bildschirm bewegen. Fahren Sie einmal über einzelne Menüpunkte hinweg. Sie sehen dann, daß die Menüpunkte, über denen der Cursor gerade steht, invers dargestellt werden. Indem Sie den Joystickknopf drücken, wählen Sie den jeweiligen Menüpunkt an. Die einzelnen Menüpunkte bedeuten folgendes:

Eingabe: Damit läßt sich ein Zeichen aus dem Zeichensatz in den Zeichenbildschirm kopieren, um es zu verändern. Nachdem Sie diesen Menüpunkt angewählt haben, erscheint unter der Codezahl ein kleiner Pfeil. Indem sie den Joystick dann hinauf- oder hinunterdrücken, läßt sich die Codezahl abwandeln. Der Pfeil ist auch nach rechts und links verschiebbar, um die Zehner- und Hunderterstellen der Codezahl zu verändern. Durch Druck auf den Joystickknopf beenden Sie die Eingabe, und das ausgewählte Zeichen erscheint im Zeichenfenster.

Nächstes Zeichen: Mit diesem Menüpunkt wählen Sie das Zeichen mit der darauffolgenden Codezahl aus und holen es vom Zeichensatz ins Zeichenfenster.

Lösche Zeichen: Das Zeichen mit der ausgewählten Codezahl wird gelöscht.

Kopiere Zeichen: Kopiert ein über die Codezahl auswählbares Zeichen an die gerade bearbeitete Stelle des Zeichensatzes.

Lösche alles: Löscht den gesamten Zeichensatz.

Intern. Zeichensatz: Hiermit kopiert man den Atari-Zeichensatz mit den internationalen Sonderzeichen in den Arbeitsspeicher des Programms.

Standardzeichen: Wie oben, nur mit dem normalem Zeichensatz.

BSAVE: Speichert einen Zeichensatz auf Diskette.

BLOAD: Lädt einen Zeichensatz von Diskette.

Directory: Zeigt das Inhaltsverzeichnis der Diskette an. Gleichzeitig läßt sich ein Zeichensatz wie bei BLOAD laden, indem der jeweilige Directory-Eintrag mit dem Cursor ausgewählt wird.

Spiegeln: Das Zeichen im Zeichenfenster wird gespiegelt, wobei zwei Spiegelachsen zur Verfügung stehen. Diese werden durch Hoch-Nieder- oder Links-Rechts-Bewegung mit dem Joystick ausgewählt.

Invers invertiert das Zeichen im Zeichenfenster. Helle Punkte werden in dunkle und dunkle Punkte in helle Punkte umgewandelt.

Verschieben: Verschiebt ein Zeichen in der angegebenen Richtung. Die Richtung wird durch den Joystick ausgewählt. Bildschirmpunkte, die aus dem Zeichenfenster hinausgeschoben werden, kommen am gegenüberliegenden Rand wieder zum Vorschein.

Graphics 0 stellt den gesamten Zeichensatz in dieser Grafikstufe dar. Ein Zeichen besteht aus acht mal acht Punkten.

Graphics 1 stellt den Zeichensatz in Grafikstufe 1 dar. Ein Zeichen besteht ebenfalls aus acht mal acht Punkten. Eine extra Anzeigemöglichkeit für Grafikstufe 2 ist nicht eingebaut, da die Zeichen dort zwar doppelt so hoch dargestellt werden, ansonsten aber genauso aufgebaut sind wie in Grafikstufe 1.

Graphics 12: In einem Zeichen lassen sich jetzt vier Farben darstellen.

```
0490:10 F7 68 AA 20 69 84 48 <D9>
0498:8A 48 AE 2F 02 CA 8E 00 <E6>
04A0:D4 AD 58 80 09 02 8D 09 <53>
04A8:D4 8D 0A D4 A2 04 BD 9B <AE>
04B0:80 9D 16 D0 CA 10 F7 68 <D4>
04B8:AA 20 69 84 48 AD 58 80 <0A>
04C0:8D 0A D4 EA 8D 09 D4 20 <02>
04C8:69 84 48 AD 58 80 09 02 <D2>
04D0:8D 0A D4 EA 8D 09 D4 20 <33>
04D8:69 84 40 A9 11 8D 6F 02 <66>
04E0:A9 FF 8D 09 D0 8D 0A D0 <B7>
04E8:8D 0B D0 8D 0C D0 A9 30 <EF>
04F0:8D 01 D0 8D 02 D0 8D 03 <AD>
04F8:D0 8D 07 D0 A9 38 8D 06 <B6>
0500:D0 A9 40 8D 05 D0 A9 48 <5E>
0508:8D 04 D0 EE 35 80 A2 00 <A3>
0510:20 A1 81 20 8C 85 A2 01 <1B>
0518:20 A1 81 20 8C 85 A2 02 <24>
0520:20 A1 81 20 8C 85 A2 03 <4D>
0528:20 A1 81 20 8C 85 A2 00 <92>
0530:20 A1 81 A9 FF A0 10 91 <88>
0538:CB C8 C0 30 D0 F9 A9 10 <58>
0540:8D 94 80 AD 38 80 F0 05 <15>
0548:A9 40 8D 94 80 A0 3C A9 <25>
0550:50 2D 94 80 20 96 85 A2 <D6>
0558:01 20 A1 81 A0 40 A9 40 <B6>
0560:2D 94 80 20 96 85 A2 02 <C1>
0568:20 A1 81 A0 44 A9 40 2D <06>
0570:94 80 20 96 85 A2 03 20 <9A>
0578:A1 81 A0 40 A9 10 2D 94 <1F>
0580:80 20 96 85 A9 40 A0 48 <A9>
0588:2D 94 80 20 96 85 CE 35 <AB>
0590:80 60 A0 7F A9 00 91 CB <12>
0598:88 10 FB 60 A2 03 91 CB <4D>
05A0:C8 CA 10 FA 60 AE 38 80 <4C>
05A8:D0 16 8E 3B 80 8E 3C 80 <D2>
05B0:AE 46 80 F0 02 49 FF 8D <F2>
05B8:3D 80 49 FF 8D 3A 80 60 <0E>
05C0:A2 03 A0 00 8C 3A 80 8C <28>
05C8:3B 80 8C 3C 80 8C 3D 80 <AC>
05D0:48 29 C0 48 49 C0 D0 06 <74>
05D8:20 19 86 18 90 1B 68 48 <71>
05E0:29 80 F0 06 20 2A 86 18 <8A>
05E8:90 0F 68 48 29 40 F0 06 <18>
05F0:20 3B 86 18 90 03 20 4C <A0>
05F8:86 68 E0 00 F0 1F 0E 3A <AD>
0600:80 0E 3A 80 0E 3B 80 0E <87>
0608:3B 80 0E 3C 80 0E 3C 80 <43>
0610:0E 3D 80 0E 3D 80 68 0A <09>
0618:0A 48 CA 10 B4 68 60 AD <1E>
0620:3D 80 09 03 8D 3D 80 AD <41>
0628:3A 80 29 FC 8D 3A 80 60 <02>
0630:AD 3C 80 09 03 8D 3C 80 <85>
0638:AD 3A 80 29 FC 8D 3A 80 <D0>
0640:60 AD 3B 80 09 03 8D 3B <AF>
0648:80 AD 3A 80 29 FC 8D 3A <3B>
0650:80 60 AD 3A 80 09 03 8D <20>
0658:3A 80 60 68 EE 35 80 68 <4E>
0660:68 20 26 84 A5 CB 85 CD <4B>
0668:A5 CC 85 CE A0 00 8C 3F <BC>
0670:80 A2 10 8E 40 80 AC 3F <21>
0678:80 B1 CD 20 9F 85 A2 00 <1B>
0680:20 A1 81 AC 40 80 AD 3A <4F>
0688:80 20 96 85 A2 01 20 A1 <CA>
0690:81 AC 40 80 AD 3B 80 20 <D3>
0698:96 85 A2 02 20 A1 81 AC <97>
06A0:40 80 AD 3C 80 20 96 85 <74>
06A8:A2 03 20 A1 81 AC 40 80 <0B>
06B0:AD 3D 80 20 96 85 AD 40 <D0>
06B8:80 18 69 04 8D 40 80 EE <4A>
06C0:3F 80 AD 3F 80 C9 08 D0 <D5>
06C8:AD CE 35 80 60 68 EE 35 <DA>
06D0:80 68 68 8D 3A 80 68 68 <85>
06D8:8D 3B 80 68 68 8D 3C 80 <CC>
06E0:68 68 8D 3D 80 AD 3A 80 <BF>
06E8:20 26 84 AC 3C 80 B1 CB <65>
06F0:A8 AE 38 80 D0 0C AE 3B <B2>
06F8:80 AD 3D 80 20 28 87 18 <1E>
0700:90 15 AD 3D 80 29 01 AE <55>
0708:3B 80 20 28 87 A8 E8 AD <71>
0710:3D 80 29 02 20 28 87 20 <90>
0718:44 87 CE 35 80 60 01 02 <B4>
0720:04 08 10 20 40 80 FE FD <2B>
0728:FB F7 EF DF BF 7F 8D 40 <86>
0730:80 98 3D 20 87 8D 41 80 <76>
0738:AD 40 80 F0 09 AD 41 80 <BD>
0740:1D 18 87 8D 41 80 AD 41 <4F>
0748:80 60 AC 3C 80 91 CB 60 <9B>
0750:EE 35 80 68 68 68 AA 09 <7C>
0758:40 A8 8A 09 80 8C 73 80 <8B>
0760:8D 76 80 8E 77 80 8E 79 <3A>
0768:80 8D 7A 80 8E 7B 80 8E <71>
0770:7C 80 8E 78 80 CE 35 80 <62>
0778:60 68 EE 35 80 68 68 20 <26>
0780:26 84 A0 07 8C 3A 80 B1 <FA>
0788:CB 20 92 87 AC 3A 80 91 <E5>
0790:CB 88 10 F0 CE 35 80 60 <CB>
0798:AE 38 80 D0 06 20 A1 87 <51>
07A0:18 90 03 20 B0 87 60 A2 <F3>
07A8:07 8D 3B 80 A9 00 4E 3B <C7>
07B0:80 2A CA 10 F9 60 A2 03 <0F>
07B8:8D 3B 80 A9 00 8D 3C 80 <81>
07C0:AD 3B 80 29 C0 0D 3C 80 <A1>
07C8:8D 3C 80 E0 00 F0 0F 0E <24>
07D0:3B 80 0E 3B 80 4E 3C 80 <F7>
07D8:4E 3C 80 CA 10 E2 AD 3C <C6>
07E0:80 60 68 EE 35 80 68 68 <56>
07E8:20 26 84 A0 07 B1 CB AE <C9>
07F0:38 80 D0 05 49 FF 18 90 <0B>
07F8:03 20 FF 87 91 CB 88 10 <AA>
0800:EC CE 35 80 60 A2 03 8D <64>
0808:3A 80 A9 00 8D 3B 80 AD <2F>
0810:3A 80 18 D8 69 40 2A 2A <4E>
0818:2A 29 03 0D 3B 80 8D 3B <A6>
0820:80 E0 00 F0 0F 0E 3B 80 <0A>
0828:0E 3B 80 0E 3A 80 0E 3A <E0>
0830:80 CA 10 DB AD 3B 80 60 <19>
0838:68 EE 35 80 68 68 8D 3A <C6>
0840:80 68 68 8D 3B 80 AD 3A <6F>
0848:80 20 26 84 68 68 8D 3C <61>
0850:80 AD 3C 80 F0 11 A0 07 <E6>
0858:B1 CB 20 65 88 91 CB 88 <AA>
0860:10 F6 CE 3C 80 D0 EA CE <48>
0868:35 80 60 AE 3B 80 D0 0C <1D>
0870:20 82 88 AE 38 80 F0 03 <AF>
0878:20 82 88 60 20 86 88 AE <F8>
0880:38 80 F0 03 20 86 88 60 <40>
0888:0A 69 00 60 4A 90 02 09 <97>
0890:80 60 68 EE 35 80 68 68 <A5>
0898:20 26 84 85 D1 A5 CB 85 <20>
08A0:D4 A5 CC 85 D5 CE 35 80 <71>
08A8:60 68 EE 35 80 68 68 85 <C9>
08B0:CC A9 00 85 CB 85 CD AD <E0>
08B8:58 80 85 CE A9 FF 8D 01 <A5>
08C0:D3 A2 04 A0 00 B1 CB 91 <1A>
08C8:CD 88 D0 F9 E6 CC E6 CE <C2>
08D0:CA D0 F2 CE 35 80 60 68 <E5>
08D8:60 00 00 00 00 00 00 00 <98>
08E0:00 00 00 00 00 00 00 00 <F0>
08E8:00 00 00 00 00 00 00 00 <F8>
08F0:00 00 00 00 00 00 00 00 <00>
08F8:00 00 00 00 00 00 00 00 <08>
0900:00 00 00 00 00 00 00 00 <12>
```

Listing. »CHAR.OBJ« wird vom Char-Maker benötigt. Bitte mit AMPEL eingeben.

Lösche Teil: Damit läßt sich ein beliebig langer Block von Zeichen aus dem Zeichensatz löschen. Mit der Codezahl werden das erste und das letzte zu löschende Zeichen ausgewählt.

Lade Teil: Aus einem auf der Diskette vorhandenem Zeichensatz wird ein beliebig langer Teil in den gerade bearbeiteten Zeichensatz geladen. Das erste und das letzte zu ladende Zeichen wird über die Codezahl ausgewählt.

LIST: Mit diesem Befehl läßt sich ein beliebiger Teil aus einem Zeichensatz in ein Basic-Programm einbauen. Das Programm fragt dann, an welcher Speicherseite es stehen soll (siehe auch Beitrag »Die Handschrift des Computers« in diesem Heft). Dann wird nach dem ersten und letzten zu übertragenden Zeichen gefragt. Wollen Sie den gesamten Zeichensatz übertragen, so geben Sie 0 und 127 ein. Anschließend folgt die Frage, ob ein zweiter Block mit Zeichen gespeichert werden soll. Diese beantwortet man mit »ja«, indem man den Joystickknopf drückt, oder mit »nein«, indem man den Joystick hinauf oder hinunter bewegt. Dann folgt die Frage nach der ersten Zeilennummer im Programm sowie nach der Schrittweite der Nummern. Zuletzt fragt das Programm nach dem Namen, den es dem File geben soll. Auf der Diskette wird dann ein ASCII-File unter dem angegebenen Namen gespeichert. Dieses wird mit dem »ENTER«-Befehl in ein Basic-Programm eingebunden. Das auf der Diskette erzeugte File enthält gleichzeitig ein Ladeprogramm, das den Zeichensatz unter Basic installiert.

Farben verändert die gerade ausgewählte Farbe.

Ein Zeichen editiert man, indem man mit dem Cursor ins Zeichenfenster fährt und dort durch Druck auf den Joystickknopf Punkte setzt oder löscht. Ein Zeichen läßt sich zusätzlich auswählen, indem Sie mit dem Cursor auf die Codezahl fahren und den Knopf drücken. Sie erhalten dann die gleiche Funktion wie beim Menüpunkt »Eingabe«.

Tritt beim Laden oder Speichern auf Diskette ein Fehler auf, so wird die Fehlernummer unter den Menüpunkten gezeigt. Um mit dem Programm fortzufahren, müssen Sie dann den Joystickknopf drücken. (Thomas Richter/hf)

Steckbrief	
Programm:	Char-Maker
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer und AMPEL
Datenträger:	Diskette

```
2 DPOKE 566,DPEEK(528)                       <OY>
5 TRAP 5:CLOSE #1:OPEN #1,4,0,"D:CHAR3.O
BJ":CLOSE #1                                 <DH>
10 GRAPHICS 0:POKE 559,0:DLO=DPEEK(560):
PAUSE 0:COL=$809B                            <WB>
15 POKE COL,0:POKE COL+1,10:POKE COL+2,0
:POKE COL+4,0                                <BJ>
20 BLOAD "D:CHAR3.OBJ":POKE 559,34           <YG>
30 POKE $8058,$A0:POKE $A000,0:MOVE $A00
```

Listing. »CHAR.BAS«, bitte mit Prüfsummer eingeben

```
0,$A001,$03FF                                   <AN>
40 U=USR($8000):MCO=3:GRAF=0:AS=32:POKE
705,$32:POKE 711,0:POKE 706,$76:POKE 707
,$1A                                            <YS>
50 POKE 756,$CC:SETCOLOR 0,8,6:SETCOLOR
4,8,0:DIM F$(20),FI$(20),H$(40),AN(16),E
N(16),ANAD(16),ENAD(16)                         <EY>
60 EXEC MENUE:YA=-1:EXEC GR0                    <SM>
65 --                                           <ZL>
70 X=PEEK($8033):Y=PEEK($8034):XH=11+18*
(X>28):POKE 77,0                                <OV>
80 IF X<8 AND Y<8 THEN GO# ZEEID                <EA>
81 IF Y=9 AND X>=5 AND X<=7 THEN X=11:Y=
2:GOTO 120                                      <GQ>
90 IF (Y<2 OR Y>11) OR (X<11 OR (X>26 AN
D X<29)) THEN IF YA>0:U=USR($8015,XA,YA,
11+5*(XA<29)):YA=-1:ENDIF :GOTO 110             <BD>
100 IF Y<>YA AND YA>-1 THEN U=USR($8015,
XH,YA,11+5*(XA<29)):U=USR($8015,XH,Y,11+
5*(XH<29)):XA=XH:YA=Y                           <DY>
101 IF Y<>YA AND YA=-1 THEN U=USR($8015,
XH,Y,11+5*(XH<29)):YA=Y:XA=XH                   <VL>
110 IF PEEK($D01F)=5 AND STICK(0)=15 THE
N EXEC FARBUM                                   <HK>
120 IF STRIG(0)=1 THEN 70                       <BU>
122 IF (X>=1 AND X<=4) AND (Y>=11 AND Y<
=14) THEN MCO=Y-11:EXEC GONG:IF GRAF<12:
MCO=3*MCO:ENDIF :EXEC PARAM:GOTO 70             <OK>
125 IF (Y<2 OR Y>11) OR (X<11 OR (X>26 A
ND X<29)) THEN EXEC GONG:GOTO 70                <DW>
130 MN=(Y-2)+10*(X>28)+1:PAUSE 0:EXEC GO
NG                                              <TK>
135 IF STICK(0)<15 THEN 70                      <AL>
140 ON MN EXEC EIN,NAE,LOE,KOP,ALO,INE,S
TA,BSA,BLO,DIR,SPI,INV,VER,GR0,GR1,GR2,D
RE,PRO,LIS,END                                  <YW>
150 POKE 77,0:PAUSE 5:GOTO 70                   <QD>
151 --                                          <JP>
152 # NEWPO:U=USR($8015,XA,YA,11+5*(XA<2
9)):XHN=11+18*(XN>28):U=USR($8015,XHN,YN
,11+5*(XHN<29)):GOTO 70                         <XK>
155 --                                          <KB>
160 # ZEEID:XA=-1:YA=-1                         <TK>
170 PAUSE 1:X=PEEK($8033):Y=PEEK($8034)         <JH>
180 IF X>7 OR Y>7 THEN YA=-1:GOTO 70            <TU>
185 IF STICK(0)=15 AND PEEK($D01F)=5 THE
N EXEC FARBUM                                   <NV>
190 IF STRIG(0)=1 THEN 170                      <ZN>
195 BW=MCO:IF GRAF<>12 AND MCO=3 THEN BW
=1                                              <JU>
196 IF GRAF=12 THEN U=USR($8006,AS,(7-X)
-1,Y,BW):GOTO 210                               <QJ>
200 U=USR($8006,AS,7-X,Y,BW)                    <AZ>
210 U=USR($8003,AS):POKE $D01F,0:XA=X:YA
=Y                                              <SK>
220 GOTO 170                                    <OK>
999 --                                          <LD>
1000 PROC MENUE:POKE 752,1:POKE 82,8:POK
E 83,39:POKE 87,0:? "{ESC CTL <}";:POSIT
ION 8,0                                         <AZ>
1010 ? "1▲▲von▲THOR-Software▲(c)▲1986/87
";                                              <MC>
1020 ? "2"                                      <KX>
1030 ? "3▲▲Eingabe▲▲▲▲▲▲▲▲▲▲▲Spiegeln"          <MX>
1040 ? "4▲▲N{CTL K}chstes▲Zeichen▲▲Inver
s"                                              <SE>
1050 ? "5▲▲L{CTL O}sche▲▲▲Zeichen▲▲Versc
hieben";                                        <KB>
1060 ? "6▲▲Kopiere▲▲Zeichen▲▲Graphics▲0"        <DM>
1070 ? "7▲▲L{CTL O}sche▲alles▲▲▲▲▲▲Graph
ics▲1"                                          <JX>
1080 ? "8▲▲Intern.▲Zeichen▲▲▲Graphics▲12
"                                               <FE>
1090 POSITION 0,8:? "87654321▲▲▲Standart
zeichen▲▲▲L{CTL O}sche▲Teil";                   <WI>
1100 ? "▲▲▲BSAVE▲▲▲▲▲▲▲▲▲▲▲▲▲Lade▲Teil"         <BZ>
1110 ? "▲▲▲BLOAD▲▲▲▲▲▲▲▲▲▲▲▲▲LIST"              <PS>
1120 ? "▲▲▲Directory▲▲▲▲▲▲▲▲▲Farben"            <MI>
1121 POSITION 0,16:? "___________________
_____________________";                         <MJ>
1130 EXEC PARAM:ENDPROC                         <TQ>
1140 PROC PARAM:FOR I=11 TO 14:POSITION
2,I:? "▲▲▲▲";:NEXT I                            <BN>
1150 POSITION 0,9:? "Code:";:VL=AS:EXEC
PRIN                                            <DK>
1160 POSITION 0,15:? "Modus:";GRAF;"▲▲";        <HW>
1170 POSITION 10,14:? "-Keine▲Fehler-"          <JK>
1180 IF GRAF=0 OR GRAF=1                        <KF>
1190 POSITION 4,11+(MCO=3):? "{ESC ESC}{
ESC CTL +}";                                    <OZ>
1200 ELSE :POSITION 2,11+MCO:? "{ESC ESC
}{ESC CTL +}";:ENDIF                            <KC>
1210 ENDPROC                                    <OJ>
1220 PROC PRIN:H$="000"                         <SJ>
1230 F$=STR$(VL):H$(4-LEN(F$))=F$:? H$;:
ENDPROC                                         <BT>
1240 PROC FARBUM:IF GRAF=0 OR GRAF=1            <WF>
1250 POSITION 4,11+(MCO=3):? "▲";               <GB>
1260 ELSE :POSITION 2,11+MCO:? "▲";:ENDI
F                                               <JC>
1270 MCO=MCO+1+2*(GRAF<>12)                     <EP>
1280 IF MCO>3 THEN MCO=0                        <UF>
1290 EXEC PARAM:EXEC GONG:ENDPROC               <PY>
1300 PROC GONG:FOR I=15 TO 0 STEP -1:SOU
ND 0,120,10,I:SOUND 1,121,10,I:PAUSE 0:N
EXT I:ENDPROC                                   <GX>
1310 PROC PLOF:POKE $8035,1:POKE 53248,0
:ENDPROC                                        <CR>
1320 PROC PLON:POKE $8035,0:ENDPROC             <MW>
1330 PROC SHURE                                 <IK>
1340 SOUND 0,60,10,10:REPEAT :UNTIL STRI
G(0)=1                                          <SE>
1350 POSITION 11,17:? "Ganz▲sicher?(Feue
rknopf!)";:YE=0:PAUSE 3                         <WP>
1360 FOR I=10 TO 0 STEP -0.2                    <QY>
1370 SOUND 0,144,10,I:SOUND 1,145,10,I          <KS>
1380 IF STRIG(0)=0 THEN YE=1:EXIT               <PE>
1390 IF STICK(0)<>15 THEN EXIT                  <JG>
1400 PAUSE 0:NEXT I:SOUND 0,0,0,0:SOUND
1,0,0,0                                         <HK>
1405 REPEAT :UNTIL STRIG(0)=1                   <WV>
1410 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:IF YE=0 THEN
 POP :POP :GOTO 70                              <MJ>
1420 ENDPROC                                    <OQ>
1430 PROC DISKIN:DPOKE 560,DLO:POKE $D40
E,$40:SETCOLOR 4,3,4:SETCOLOR 2,0,10:SET
COLOR 1,0,0                                     <FA>
1450 POKE 82,0:POSITION 0,19                    <TS>
1470 ? "▲▲▲▲▲Diskettenzugriff!▲Bitte▲▲▲"        <SG>
1480 ? "▲▲▲▲▲einen▲Augenblick▲Geduld▲▲!"        <GZ>
1500 POKE 82,8:POSITION 8,2:EXEC PLOF           <BC>
1510 ENDPROC                                    <OP>
1520 PROC DISKOU:U=USR($8000):EXEC PLON:
SETCOLOR 4,8,0:SETCOLOR 0,8,6                   <EE>
1521 POSITION 0,18:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT DEL}
{ESC SHIFT DEL}{ESC SHIFT DEL}{ESC SHIFT
 DEL}{ESC SHIFT DEL}{ESC SHIFT DEL}";:U=
USR($8003,AS):ENDPROC                           <XO>
1530 PROC FILEIN:POKE 752,0                     <BJ>
1540 TRAP 1540:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";           <WC>
1550 POSITION 11,17:? "Filename▲D1:";           <QF>
1560 INPUT "{ESC CTL *}{ESC CTL +}",F$:F
I$="D:":FI$(3)=F$:POSITION 0,17:? "{ESC
SHIFT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";
:TRAP 40000:POKE 752,1:? "{ESC CTL *}{ES
C CTL +}";:ENDPROC                              <LH>
1570 PROC BEGENI:ASS=AS                         <FR>
1580 POSITION 11,17:? "Erstes▲▲Zeichen"         <CH>
1590 EXEC EIN:AN=AS                             <SP>
1600 POSITION 11,17:? "Letztes▲Zeichen"         <UY>
1610 EXEC EIN:EN=AS:AS=ASS:POSITION 5,9:
VL=AS:EXEC PRIN                                 <YT>
1620 ANAD=USR($8018,AN):ANI=PEEK(209)           <AG>
1630 ENAD=USR($8018,EN):ENI=PEEK(209)           <IJ>
1640 POSITION 11,17:? "{ESC SHIFT >}{ESC
 SHIFT DEL}{ESC SHIFT DEL}";:ENDPROC            <ME>
2000 PROC EIN:EXEC PLOF                         <QS>
2010 H$=STR$(AS):X=7:FI$="000":FI$(4-LEN
```




```
(H$))=H$:PAUSE 2                              <CU>
2020 PAUSE 2:ST=STICK(0):W=ASC(FI$(X-4))
-48:SOUND 0,0,0,0                             <NZ>
2030 IF ST=11 AND X>5 THEN X=X-1:W=ASC(F
I$(X-4))-48                                   <IJ>
2040 IF ST=7 AND X<7 THEN X=X+1:W=ASC(FI
$(X-4))-48                                    <AZ>
2050 IF ST=13 THEN W=W-1:IF W=-1 THEN W=
9                                             <RJ>
2060 IF ST=14 THEN W=W+1:IF W=10 THEN W=
0                                             <FO>
2070 FI$(X-4,X-4)=CHR$(W+48):POSITION 5,
9:? FI$                                       <XR>
2080 POSITION X-1,10:? "▲{ESC ESC}{ESC C
TL -}▲";:IF ST<>15 THEN SOUND 0,60,10,10 <FG>
2090 IF STRIG(0)=1 THEN 2020                  <DG>
2091 AS=VAL(FI$):IF AS>127 THEN EXEC GON
G:GOTO 2020                                   <QU>
2100 U=USR($8003,AS):VL=AS:POSITION 5,10
:? "▲▲▲▲";:SOUND 0,0,0,0:EXEC PLON:ENDPR
OC                                            <DR>
2110 PROC NAE:AS=AS+1:IF AS=128 THEN AS=
0                                             <NK>
2120 U=USR($8003,AS):POSITION 5,9:VL=AS:
EXEC PRIN:ENDPROC                             <WY>
2130 PROC LOE:EXEC SHURE                      <SV>
2140 U=USR($8018,AS)                          <VU>
2150 FOR I=U TO U+7:POKE I,0:NEXT I:U=US
R($8003,AS):ENDPROC                           <MA>
2160 PROC KOP:POSITION 11,17:? "Welches▲
Zeichen▲?"                                    <BQ>
2170 ASS=AS:EXEC EIN                          <VL>
2180 U=USR($8018,ASS):V=USR($8018,AS)         <DU>
2190 MOVE V,U,8:AS=ASS:U=USR($8003,AS):P
OSITION 0,17:? "{ESC SHIFT >}{ESC SHIFT
DEL}{ESC SHIFT DEL}";:POSITION 5,9:VL=AS
:EXEC PRIN:ENDPROC                            <TM>
2200 PROC ALO:EXEC SHURE:POKE $A000,0:MO
VE $A000,$A001,$03FF                          <NG>
2210 U=USR($8003,AS):ENDPROC                  <FR>
2220 PROC INE:EXEC SHURE:U=USR($801B,$CC
):U=USR($8003,AS):ENDPROC                     <VL>
2230 PROC STA:EXEC SHURE:U=USR($801B,$E0
):U=USR($8003,AS):ENDPROC                     <AI>
2240 PROC BSA:EXEC FILEIN:POKE 752,0          <MQ>
2250 TRAP 2250:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";         <VO>
2260 POSITION 11,17:? "In▲welche▲Page?▲9
6{ESC CTL +}{ESC CTL +}";                     <NG>
2270 INPUT "{ESC CTL *}{ESC CTL +}",PG:P
OKE 752,1                                     <EN>
2280 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";                   <ES>
2290 EXEC DISKIN                              <SS>
2300 TRAP #ERRO:OPEN #2,8,0,FI$               <FO>
2310 PUT #2,$FF,$FF,0,PG,$FF,PG+3             <YU>
2320 BPUT #2,$A000,$0400                      <XU>
2330 CLOSE #2:EXEC DISKOU:TRAP 40000:END
PROC                                          <IJ>
2340 PROC BLO:EXEC FILEIN                     <QW>
2350 EXEC DISKIN:TRAP #ERRO                   <DO>
2360 OPEN #2,4,0,FI$:GET #2,A,A               <YJ>
2370 IF A+A<>510 THEN 2400                    <QC>
2380 GET #2,A,A,A,A                           <DX>
2390 BGET #2,$A000,$0400                      <RG>
2400 CLOSE #2:TRAP 40000:EXEC DISKOU:U=U
SR($8003,AS):ENDPROC                          <AH>
2500 # ERRO:TRAP #ERRO:CLOSE #2:EXEC DIS
KOU                                           <TL>
2510 SOUND 0,12,12,15:SOUND 1,15,12,15        <OX>
2520 PAUSE 40:SOUND 0,0,0,0:SOUND 1,0,0,
0:POSITION 10,14:? "▲▲Error▲-▲";ERR;"▲▲▲ <QP>
2530 REPEAT :UNTIL STRIG(0)=0                 <VK>
2540 POSITION 10,14:? "▲▲▲▲-Keine▲Fehler
-";:POP :POP :GOTO 70                         <TP>
2550 PROC DIR:EXEC DISKIN                     <BU>
2560 TRAP #ERRO:OPEN #2,6,0,"D:*.*"           <NO>
2570 TRAP 2620:C=0:POKE 82,8:? "{ESC CTL
*}{ESC CTL +}";:BI=DPEEK(88)                  <HH>
2580 FOR I=0 TO 11:AD=BI+I*40+8:POKE AD,
0:MOVE AD,AD+1,31:NEXT I:FOR I=690 TO 69
3:POKE I,255:NEXT I                           <UB>
2585 POKE 93,0:E=1                            <HA>
2590 TRAP 2620:INPUT #2,F$:FI$="":IF F$(
2,2)<>"▲" THEN E=0                            <YP>
2591 TRAP 2592:INPUT #2,FI$                   <AH>
2592 IF F$(2,2)="▲" THEN F$=F$(3)             <JU>
2593 IF FI$<>"" THEN IF FI$(2,2)="▲":FI$
=FI$(3):ELSE :POSITION 8,7:? "{ESC SHIFT
DEL}":E=0:ENDIF                               <YZ>
2600 POSITION 8,C:? F$;"▲";FI$;:C=C+1:IF
C>=7 AND E THEN EXEC WTFK:C=0:GOTO 2580 <VO>
2610 GOTO 2590                                <UK>
2620 TRAP 40000:EXEC WTFK:EXEC DISKOU:EX
EC MENUE:CLOSE #2:YA=-1:ENDPROC               <CS>
2630 PROC WTFK:EXEC DISKOU:POSITION 2,17
:? "Bitte▲ausw{CTL K}hlen▲oder▲START▲dr{
CTL J}cken!":EXEC DISKCHOI                    <OX>
2640 REPEAT :UNTIL PEEK($D01F)=7:POSITIO
N 2,17:? "{ESC SHIFT >}{ESC SHIFT DEL}{E
SC SHIFT DEL}{ESC SHIFT >}";:EXEC DISKIN
:ENDPROC                                      <AP>
2650 PROC SPI:POSITION 11,17:EXEC PLOF        <EG>
2660 ? "Spiegeln:▲{ESC ESC}{ESC CTL -}{E
SC ESC}{ESC CTL =}":SP=0                      <CJ>
2670 ST=STICK(0)                              <JI>
2680 IF ST=13 OR ST=14 THEN SP=0:POSITIO
N 21,17:? "{ESC ESC}{ESC CTL -}{ESC ESC}
{ESC CTL =}":EXEC GONG                        <FD>
2690 IF ST=7 OR ST=11 THEN SP=1:POSITION
21,17:? "{ESC ESC}{ESC CTL +}{ESC ESC}{
ESC CTL *}":EXEC GONG                         <XK>
2700 IF STRIG(0)=1 THEN 2670                  <NY>
2710 IF SP=1 THEN 2750                        <NM>
2720 U=USR($8018,AS):MOVE U,$0680,8:FOR
I=0 TO 7:POKE U+I,PEEK($0680+(7-I)):NEXT
I                                             <SZ>
2730 U=USR($8003,AS):GOTO 2760                <JO>
2750 U=USR($800C,AS):U=USR($8003,AS)          <DB>
2760 EXEC PLON:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";:EN
DPROC                                         <RK>
2770 PROC INV:U=USR($800F,AS):U=USR($800
3,AS):ENDPROC                                 <XC>
2780 PROC VER:SP=0:EXEC PLOF                  <VH>
2790 POSITION 11,17:? "Verschieben:▲{ESC
ESC}{ESC CTL -}"                              <AE>
2800 ST=STICK(0)                              <IR>
2810 IF ST=14 THEN SP=0:POSITION 24,17:?
"{ESC ESC}{ESC CTL -}":EXEC GONG              <VU>
2811 IF ST=13 THEN SP=1:POSITION 24,17:?
"{ESC ESC}{ESC CTL =}":EXEC GONG              <YA>
2820 IF ST=7 THEN SP=2:POSITION 24,17:?
"{ESC ESC}{ESC CTL *}":EXEC GONG              <KQ>
2821 IF ST=11 THEN SP=3:POSITION 24,17:?
"{ESC ESC}{ESC CTL +}":EXEC GONG              <AP>
2830 IF STRIG(0)=1 THEN 2800                  <JC>
2840 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";                   <ES>
2850 POSITION 11,17:? "Wie▲oft▲?▲1":OFT=
1                                             <VG>
2860 PAUSE 3:ST=STICK(0):SOUND 0,0,0,0        <UI>
2870 IF ST=13 AND OFT>1 THEN OFT=OFT-1        <JU>
2880 IF ST=14 AND OFT<7 THEN OFT=OFT+1        <LE>
2890 IF ST<>15 THEN SOUND 0,60,10,10          <LO>
2900 POSITION 21,17:? CHR$(OFT+48);           <NW>
2910 IF STRIG(0)=1 THEN 2860                  <PE>
2920 ON SP+1 EXEC UP,DO,LR,LR                 <WR>
2930 U=USR($8003,AS):EXEC PLON:POSITION
0,17:? "{ESC SHIFT >}{ESC SHIFT DEL}{ESC
SHIFT DEL}";:ENDPROC                          <EU>
2940 PROC UP:U=USR($8018,AS):FOR I=1 TO
OFT:A=PEEK(U):MOVE U+1,U,7:POKE U+7,A:NE
XT I:ENDPROC                                  <ZD>
2950 PROC DO:U=USR($8018,AS):FOR I=1 TO
OFT                                           <RO>
2960 A=PEEK(U+7):-MOVE U,U+1,7:POKE U,A:
NEXT I:ENDPROC                                <HA>
```

Listing. »CHAR.BAS« (Fortsetzung)

```
2970 PROC LR:SW=1                              <AA>
2980 IF SP=3 THEN SW=0                         <SO>
2990 U=USR($8012,AS,SW,OFT):ENDPROC            <WV>
3000 PROC GR0                                  <IZ>
3010 U=USR($8009,2):POKE $8038,0:GRAF=0:
POKE COL+1,PEEK(707):POKE COL+2,PEEK(711
):POKE COL+4,PEEK(711)                         <UH>
3015 EXEC CLFAU:ENDPROC                        <SJ>
3020 PROC GR1                                  <JS>
3030 U=USR($8009,6):POKE $8038,0:GRAF=1        <WS>
3040 FOR I=COL TO COL+3:POKE I,PEEK(707)
:NEXT I:POKE COL+4,PEEK(711):EXEC CLFAU:
ENDPROC                                        <TU>
3050 PROC GR2                                  <KO>
3060 U=USR($8009,4):POKE $8038,1:GRAF=12       <VV>
3070 FOR I=0 TO 2:POKE COL+I,PEEK(705+I)
:NEXT I:POKE COL+3,$C8:POKE COL+4,PEEK(7
11):EXEC CLFAU                                 <WK>
3071 POKE $8049,255:ENDPROC                    <MC>
3080 PROC CLFAU:U=USR($8000)                   <DS>
3090 FOR I=11 TO 14                            <RR>
3100 IF I<13 THEN POSITION 4,I:? "▲";          <TM>
3110 POSITION 2,I:? "▲";                       <AX>
3120 NEXT I:IF (MCO=1 OR MCO=2) AND GRAF
<>12 THEN MCO=3                                <MQ>
3130 EXEC PARAM:PAUSE 0:U=USR($8003,AS):
PAUSE 0:ENDPROC                                <CP>
3140 PROC DRE                                  <ST>
3150 EXEC SHURE:EXEC BEGENI                    <AR>
3160 POKE ANAD,0:MOVE ANAD,ANAD+1,(ENAD-
ANAD)-1+7:U=USR($8003,AS):ENDPROC              <OA>
3170 PROC PRO:EXEC BEGENI:EXEC FILEIN          <BB>
3180 EXEC DISKIN:TRAP #ERRO                    <DU>
3190 OPEN #2,4,0,FI$                           <LG>
3200 GET #2,A,B:IF A+B<>510 THEN 3300          <UE>
3210 GET #2,A,B,C,D                            <HD>
3220 BGET #2,$A800,$0400                       <VI>
3230 MOVE ANAD+$0800,ANAD,ENAD-ANAD+7          <TF>
3300 CLOSE #2:EXEC DISKOU                      <FT>
3310 TRAP 40000:U=USR($8003,AS):ENDPROC        <YD>
3320 PROC LIS                                  <YX>
3330 POKE 752,0:POSITION 0,17:? "{ESC SH
IFT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";:P
OSITION 11,17:? "Ab▲welcher▲Page?▲96{ESC
 CTL +}{ESC CTL +}";                           <HJ>
3340 TRAP 3330:INPUT "{ESC CTL *}{ESC CT
L +}",BEG                                      <HI>
3350 BEAD=BEG*256:POKE 752,1:? "{ESC CTL
 *}{ESC CTL +}";:K=2                           <RW>
3360 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:TRAP 40000         <DM>
3365 IF PEEK($D01F)=6 THEN ANAD(K-2)=32:
ENAD(K-2)=127:GOTO 3410                        <VA>
3370 EXEC BEGENI:ANAD(K-2)=ANAD:ENAD(K-2
)=ENAD:AN(K-2)=ANI:EN(K-2)=ENI:IF ENAD<A
NAD THEN 3370                                  <NC>
3380 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:POSITION 0,1
7:? K;"ten▲Block▲eingeben?(Knopf=ja,{ESC
 ESC}{ESC CTL -}{ESC ESC}{ESC CTL =}=nei
n)";:EXEC PLOF                                 <XM>
3390 IF STRIG(0)=0 AND K<18 THEN K=K+1:G
OTO 3360                                       <PJ>
3400 IF STICK(0)=15 AND K<18 THEN 3390         <QV>
3410 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:TRAP 3410:PO
KE 752,0                                       <AR>
3420 POSITION 11,17:? "Startzeilennummer
:";:INPUT "{ESC CTL *}{ESC CTL +}",ZN          <ON>
3430 TRAP 3430:POSITION 0,17:? "{ESC SHI
FT >}{ESC SHIFT DEL}{ESC SHIFT DEL}";          <VW>
3440 POSITION 11,17:? "Schrittweite:10{E
SC CTL +}{ESC CTL +}";                         <HU>
3450 INPUT "{ESC CTL *}{ESC CTL +}",ZS:T
RAP 40000                                      <TI>
3460 EXEC FILEIN                               <MR>
3470 EXEC DISKIN:TRAP #ERRO                    <DX>
3480 OPEN #2,8,0,FI$                           <NN>
3490 ? #2;ZN;"BA=";BEAD;":RESTORE";ZN:ZO
=ZN:EXEC INC                                   <XT>
3500 ? #2;ZN;"READ▲A:IF▲A<0▲THEN▲";ZO+ZS
*3:EXEC INC                                    <ZO>
3510 ? #2;ZN;"POKE▲AD,A:AD=AD+1:GOTO▲";Z
O+ZS:EXEC INC                                  <AG>
3520 ? #2;ZN;"IF▲A=-1000▲THEN▲RETURN":EX
EC INC                                         <SD>
3530 ? #2;ZN;"AD=BA+8*INT(ABS(A)):GOTO";
ZO+ZS:EXEC INC                                 <UA>
3540 FOR J=2 TO K                              <OF>
3550 ? #2;ZN;"DATA▲-";AN(J-2);".1":EXEC
INC                                            <RO>
3560 AD=ANAD(J-2):FOR L=AN(J-2) TO EN(J-
2)                                             <AU>
3570 ? #2;ZN;"DATA";:EXEC INC                  <IW>
3580 FOR I=0 TO 7                              <AH>
3590 ? #2;PEEK(AD);:AD=AD+1                    <PP>
3600 IF I<7 THEN ? #2;",";                     <RG>
3610 IF I=7 THEN ? #2                          <NL>
3620 NEXT I:NEXT L:NEXT J                      <PW>
3625 ? #2;ZN;"DATA▲-1000"                      <IV>
3630 CLOSE #2:EXEC DISKOU:TRAP 40000:END
PROC                                           <IQ>
3640 PROC INC:ZN=ZN+ZS:ENDPROC                 <ZS>
3650 PROC END:POSITION 11,17                   <QG>
3660 ? "Helligkeit:{ESC ESC}{ESC CTL -}{
ESC ESC}{ESC CTL =}▲Farbe:{ESC ESC}{ESC
CTL +}{ESC ESC}{ESC CTL *}"                    <MY>
3670 IF MCO=0 THEN REG=COL-2                   <BH>
3680 IF MCO>0 AND GRAF=12 THEN REG=704+M
CO                                             <XM>
3690 IF MCO>0 AND GRAF<>12 THEN REG=707         <OZ>
3700 EXEC PLOF                                 <HT>
3710 W=PEEK(REG)                               <TM>
3720 PAUSE 2:ST=STICK(0)                       <DG>
3730 IF ST&1=0 AND (W MOD 16)>0 THEN W=W
-1                                             <SL>
3740 IF ST&2=0 AND (W MOD 16)<15 THEN W=
W+1                                            <CR>
3750 IF ST&4=0 THEN W=W-16:IF W<0 THEN W
=W+256                                         <PS>
3760 IF ST&8=0 THEN W=W+16:IF W>255 THEN
 W=W-256                                       <SQ>
3770 POKE REG,W                                <PA>
3780 IF STRIG(0)=1 THEN 3720                   <LR>
3790 IF REG=COL-2 THEN POKE 711,W              <AA>
3800 IF GRAF=12 THEN EXEC GR2                  <HL>
3810 IF GRAF=0 THEN EXEC GR0                   <TP>
3820 IF GRAF=1 THEN EXEC GR1                   <VI>
3830 POSITION 0,17:? "{ESC SHIFT >}{ESC
SHIFT DEL}{ESC SHIFT DEL}";:EXEC PLON:EN
DPROC                                          <SW>
3840 PROC DISKCHOI:YA=-1:BI=DPEEK(88)          <XB>
3850 IF PEEK($D01F)=6 THEN ENDPROC             <DD>
3855 IF STRIG(0)=0 THEN IF H=0 AND STICK
(0)=15 AND NOT (X<8 OR Y>6 OR (X>18 AND
X<24) OR X>34) THEN 3920                       <ZW>
3860 X=PEEK($8033):Y=PEEK($8034)               <AE>
3865 H=0:XH=8+16*(X>23):W=(PEEK(XH+40*Y+
BI))&127:IF W<33 OR W>59 THEN H=1              <TT>
3870 IF X<8 OR Y>6 OR (X>18 AND X<24) OR
 H OR X>34 THEN IF YA>=0:U=USR($8015,XA,
YA,11):YA=-1:ENDIF :GOTO 3850                  <YS>
3880 IF XA=XH AND Y=YA THEN 3850               <TF>
3890 IF YA>=0 THEN U=USR($8015,XA,YA,11)       <JW>
3900 XA=XH:YA=Y:U=USR($8015,XA,YA,11)          <SM>
3910 IF STRIG(0)=1 THEN 3850                   <PD>
3920 FI$="▲▲▲▲▲▲▲▲▲▲▲"                         <OD>
3930 BD=BI+XH+Y*40                             <BK>
3940 MOVE BD,ADR(FI$),11                       <YE>
3950 BD=ADR(FI$):FOR I=BD TO BD+11             <ZY>
3960 W=PEEK(I):W=(W&127)+32:POKE I,W           <FL>
3970 NEXT I                                    <GE>
3980 FOR I=1 TO 8                              <BS>
3990 IF FI$(I,I)="▲" THEN EXIT                 <UT>
4000 NEXT I                                    <ES>
4010 F$(1,I-1)=FI$:F$(I,I)=".":F$(I+1)=F
I$(9,11):CLOSE #2                              <TN>
4020 EXEC GONG:FI$="D:":FI$(3)=F$:POP :P
OP :EXEC MENUE:YA=-1:GOTO 2350                 <FC>
```

Listing. »CHAR.BAS« (Schluß)

Ein toller Malkasten

Viele Firmen bringen Malprogramme auf den Markt, die sich nur in wenigen Details voneinander unterscheiden. Unser Malprogramm »Grafik-Draw« vereint viele Funktionen und besticht mit seinen vielfältigen Grafikstufen.

Malprogramme für die Atari 8-Bit-Computer sind genügend im Angebot. Von Datasoft gibt es den »Micropainter«, Atari und Koala-Soft haben das »Atari-Artist«-Programm beziehungsweise den »Koala-Painter« herausgebracht, und ein weiteres amerikanisches Programm namens »Fun with Art« schaltet auf Wunsch in jeder Zeile über einen Display-List-Interrupt die Farben um. Ganz zu schweigen von den vielen anderen Malprogrammen kleinerer Softwarefirmen. Warum sollte man also ein weiteres derartiges Programm veröffentlichen?

Zum einen haben die professionellen Programme den Nachteil, daß sie nur für viel Geld zu bekommen sind. Zum anderen unterstützen sie nicht die vielfältigen Grafikstufen der XL/XE-Computer. Grund genug, ein Programm zu veröffentlichen, das die meisten Grafikstufen berücksichtigt, eine umfangreiche Bibliothek an Malfunktionen besitzt, sowie obendrein noch eine Hardcopy-Routine aufweist, mit der ein Bild bequem und schnell auf einem CP-80-Drucker von Seikosha ausgedruckt wird. Wenn Sie nur eine Zeile im Programm ändern, arbeitet es auch mit anderen Druckern zusammen.

Das Programm »Grafik-Draw« besteht insgesamt aus drei Listings. Listing 1 ist das eigentliche Malprogramm. Programmteile, die zeitkritische Aufgaben übernehmen (Listing 2 und 3), sind in Maschinensprache programmiert und werden vom Hauptprogramm nach dessen Start von der Diskette dazugeladen. Diese Listings müssen mit dem AMPEL-Programm eingegeben und unter den Namen »UPADSTK.BIN« und »UKIPDRK.BIN« auf der gleichen Diskette wie das Hauptprogramm gespeichert werden.

Nach dem Start von Grafik-Draw erscheint auf dem Bildschirm das Hauptmenü. Der leuchtende Balken wählt die einzelnen Menüpunkte aus. Er wird mit einem Joystick oder einem Grafik-Tablett wie der Atari-Maltafel bewegt, die in Port 1 anzuschließen sind. Um zwischen dem Joystick oder der Maltafel als Eingabemedium zu entscheiden, geht man auf den Menüpunkt »Zeichengerät« und schaltet dann auf Knopfdruck zwischen den beiden Eingabegeräten hin und her. Das gerade gültige Zeichengerät wird rechts neben dem Menüpunkt angezeigt.

Das Programm stellt elf Zeichen sowie sechs weitere Funktionen zur Verfügung. Wurde eine Zeichenfunktion angewählt, so schaltet der Computer vom Menü zum gezeichneten Bild. Dort wird mit der Taste <F> die Malfarbe festgelegt. Zurück ins Menü kommt man mit der <SPACE>-Taste. Folgende Zeichenfunktionen stehen zur Auswahl:

Zeichnen: Freihändiges Zeichnen wird mit dieser Funktion unterstützt. Die Linie auf dem Bildschirm folgt direkt dem Fadenkreuz.

Linie: Beim ersten Druck auf den Joystick/Touch-Tablet-Knopf wird der Anfangspunkt, mit dem zweiten Knopfdruck der Endpunkt einer zu ziehenden Linie festgelegt.

Linie folgend: Wie »Linie«, nur daß der Endpunkt einer ersten Linie zugleich der Anfangspunkt einer zweiten Linie ist.

Kreis: Der erste Knopfdruck legt den Mittelpunkt eines zu zeichnenden Kreises fest. Durch Bewegen des Fadenkreuzes bestimmt man den Durchmesser des Kreises.

Ellipse: Wie Kreis, nur daß hierbei eine Ellipse gezeichnet wird, die in der Horizontalen und Vertikalen unterschiedliche Ausdehnungen hat.

Rechteck: Der erste Knopfdruck bestimmt den obersten linken Eckpunkt eines Rechtecks, der zweite Knopfdruck den untersten rechten Eckpunkt. Durch diese beiden Punkte wird ein Rechteck gezeichnet.

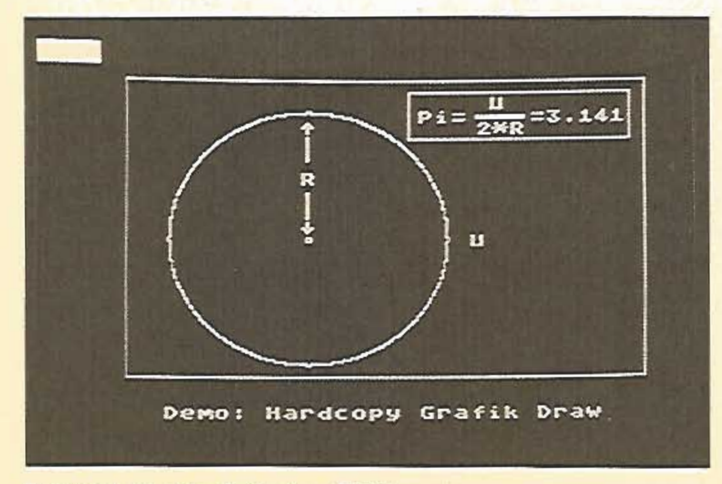
Rechteck gef.: Wie »Rechteck«. Anschließend wird das Rechteck ausgefüllt.

Ausfüllen: Malt eine beliebige schwarze Fläche wird in der gewählten Farbe aus.

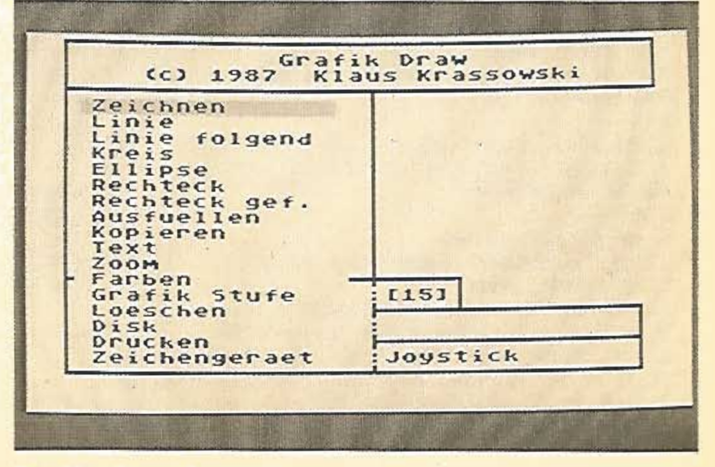
Kopieren: Diese Funktion kopiert einen beliebig großen rechteckigen Bildschirmausschnitt in einen anderen Teil des Bildes. Läßt man den Joystick/Maltafelknopf während der Bewegung des Ausschnitts gedrückt, wird mit dem Ausschnitt gezeichnet.

Text: Maximal 20 Zeichen lassen sich eingeben. Mit dem Joystick positioniert man dann den Text auf dem Bildschirm. Läßt man den Joystickknopf gedrückt, wird mit dem Text wie bei »Kopieren« auf Bildschirm gezeichnet.

Zoom: Mit dem Fadenkreuz wird ein rechteckiger Bildschirmausschnitt bestimmt, der über den ganzen Bildschirm vergrößert wird. In der Vergrößerung läßt sich nach Belieben zeichnen.



Beispiel einer technischen Zeichnung



Das Menü von Grafik-Draw



Die folgenden sechs Menüpunkte sprechen die restlichen Funktionen des Programms an:

Farben: Auf dem Bildschirm erscheinen vier Farbbalken. Neben dem obersten Balken befindet sich ein Pfeil, der die gerade gewählte Farbe markiert. Mit <START> wählt man eine andere Farbe an. <SELECT> verändert die Helligkeit der gewählten Farbe und <OPTION> variiert den Farbton. Durch <ESC> erfolgt die Übernahme der Veränderungen und die Rückkehr in den Zeichenmodus.

Grafik Stufe stellt die Auflösung des Bildes ein. Die einzelnen Stufen entsprechen denen in Atari-Basic. Zur Auswahl stehen Stufe 5, 7, 8, 9, 10, 11 und 15. Die gewählte Stufe wird rechts neben dem Menüpunkt angezeigt. Stufe 5 bietet 80 mal 48 Punkte, wobei vier Farben frei zu definieren sind. In Stufe 7 erhöht sich die Auflösung auf 160 mal 96 Bildschirmpunkte. Die höchste Auflösung bietet Stufe 8 mit 320 mal 192 Punkten. Hierbei steht aber nur noch eine Farbe in zwei Helligkeiten zur Verfügung. Stufe 9, 10 und 11 sind Unterformen von Stufe 8. Hier werden jeweils vier nebeneinanderliegende Punkte zu einem zusammengefaßt. Die Auflösung beträgt in diesen Stufen 80 mal 192. In Stufe 9 sind 16 Helligkeiten in einer Farbe verfügbar, und in Stufe 11 stehen alle 16 Farbschattierungen einer Helligkeitsstufe zur Auswahl. In Stufe 10 sind neun Farben wählbar, wobei jeder eine von insgesamt 256 Farben zugeordnet wird. In dieser so gut wie überhaupt nicht genutzten Grafikstufe des Ataris lassen sich sehr ansprechende Bilder zeichnen.

Löschen: Nach einer Sicherheitsabfrage wird das Bild gelöscht.

Disk: Das Programm verzweigt ins Disketten-Untermenü.

- <D> zeigt das Directory von Laufwerk 1 an.
- <L> lädt ein Bild von der Diskette. Dabei wird je nach Bild automatisch auf die richtige Grafikstufe umgeschaltet und die richtigen Farben gewählt. Bilder vom Atari-Artist Programm lassen sich nur dann laden, wenn sie mit <INSERT> gespeichert wurden. In diesem Fall legt der Atari-Artist die Bilder nicht in komprimierter Form auf Diskette ab.
- <S> speichert ein Bild auf Diskette.
- <ESC> kehrt ins Menü zurück.

Drucken erzeugt eine Hardcopy auf einem CP-80-Drucker. Dabei stehen zwei Vergrößerungsstufen zur Auswahl.

- <1> produziert eine Hardcopy über die halbe Breite einer DIN-A4-Seite.
- <2> druckt ein Bild über eine volle DIN-A4-Seite aus.

Bei einer Hardcopy über die halbe Breite des Blattes werden Sie zusätzlich gefragt, ob Sie den Ausdruck in der Mitte des Bildes oder nach links oder rechts versetzt haben wollen. Der Ausdruck paßt sich dann entsprechend dieser Eingabe an. Auf diese Weise lassen sich zum Beispiel doppelseitige Einladungen drucken.

Mit diesem Menüpunkt läßt sich ein Bild normal oder invers aufs Blatt nadeln. Farben werden automatisch in Grauwerte umgewandelt. Der Menüpunkt »Drucken« funktioniert nicht mit den Grafikstufen 5 und 7.

Falls Sie einen anderen Drucker als den CP-80 von Seikosha besitzen, läßt sich die Hardcopy-Routine folgendermaßen anpassen: In Zeile 10230 des Turbo-Basic-Programms steht der Drucker-Steuercode zum Drucken von Bitmustern. In diese Zeile müssen Sie die Steuercodes Ihres eigenen Druckers eintragen. Für einen Epson kompatiblen Drucker muß die Zeile wie folgt aussehen:

```
10230 GR$=CHR$(27):GR$(%2)="K":GR$(%3)=CHR$(192):
GR$(4)=CHR$(0)
```

Das Programm belegt fast den ganzen Speicher des Computers. Weitere umfangreiche Funktionen lassen sich also nicht integrieren.

Wir hoffen, daß Sie mit Grafik-Draw jetzt nicht mehr ständig viele verschiedene Malprogramme zur Hand haben müssen, um in verschiedenen Grafikstufen zeichnen zu können. Viel Spaß mit dem Programm!

(Klaus Krassowski/hf)

Steckbrief	
Programm:	Grafik-Draw
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer und AMPEL
Datenträger:	Diskette

```
10 --                                          <YW>
11 REM                                         <JQ>
12 REM Graphic Draw                            <AM>
13 REM (c) 1987  Klaus Krassowski              <BE>
15 REM                                         <JY>
49 --                                          <ZR>
500 GOSUB 10000:GOSUB 8000                     <DK>
600 BLOAD "D:UPADSTK.BIN"                      <HO>
610 Z=USR($9800,ADR(PM$),NR)                   <JK>
620 POKE $0601,TEIL                            <YG>
700 BLOAD "D:UKIPDRK.BIN":DKAD=$9500           <BT>
800 GOTO 7100                                  <MG>
999 --                                         <LD>
1500 PROC BUT:REPEAT :IF JTF:A=STICK(%1)
-14:ELSE :A=STRIG(%0):ENDIF :UNTIL A=%1:
PAUSE 4:ENDPROC                                <NB>
1699 --                                        <YB>
1700 PROC CRSGR8                               <PG>
1720 IF MODE=8 THEN POKE 87,8                  <XD>
1735 TRAP 1800                                 <QF>
1740 LOCATE X-1,Y,CR1:LOCATE X,Y-1,CR2:L
OCATE X,Y+1,CR3:LOCATE X+1,Y,CR4               <OV>
1760 COLOR 1-CR1:PLOT X-1,Y:COLOR 1-CR2:
PLOT X,Y-1:COLOR 1-CR3:PLOT X,Y+1:COLOR
1-CR4:PLOT X+1,Y:PAUSE 3                       <YT>
1780 COLOR CR1:PLOT X-1,Y:COLOR CR2:PLOT
 X,Y-1:COLOR CR3:PLOT X,Y+1:COLOR CR4:PL
OT X+1,Y                                       <PJ>
1800 COLOR COL:PAUSE 1                         <KG>
1820 IF MODE=8 THEN POKE 87,15:ENDIF           <JH>
1899 ENDPROC                                   <RD>
1999 REM _ FUNKTIONEN                          <MK>
2000 IF A=33 THEN 7000                         <QF>
2040 IF A=56 THEN GOSUB 12000                  <MD>
2060 FKTFG=%0:IF A=28 OR A=42 OR A=45 OR
 A=63 THEN 2090                                <WT>
2080 POKE N764,N255                            <DN>
2090 GOTO 2702                                 <SJ>
2500 REM ___ HAUPT                             <JM>
2600 FF=%0:CPF=%0:COL=ACOL:COLOR COL:POK
E N764,N255:GOSUB 12000                        <UO>
2620 IF GRS=10 THEN POKE 623,161               <AS>
2630 POKE $060F,%1                             <CI>
2700 # HAUPT                                   <JX>
2701 FKTFG=%1:A=PEEK(764):IF A<>N255 THE
N 2000                                         <IY>
2702 IF STICK(%1)=7 THEN 7000                  <EZ>
2703 A=PEEK(53279):IF A=%3 THEN JTF=JTF=
%0:POKE $0615,JTF:DPOKE $CB,X:POKE $CD,Y
:GOSUB 9800:FKTFG=%0                           <HZ>
2704 IF A=5 THEN JTMP=JTMP=%0:POKE $060D
,JTMP:GOSUB 9800:FKTFG=%0                      <LP>
2705 IF JTMP THEN POKE $060D,JTMP              <TG>
2710 IF JTF:P1=(DPEEK($0606) DIV TEIL-14
```

```
) DIV (APIX/%2):Y=(213-(DPEEK($0608) DIV
 TEIL)) DIV ASLI                              <QN>
2715 STR=STICK(%1)-14:X=P1:IF P1>212 THE
N X=XA:Y=YA                                   <MC>
2720 X=X*(X>%0):Y=Y*(Y>%0)                    <QH>
2721 IF X>XMAX THEN X=XMAX                    <KW>
2722 IF Y>YMAX THEN Y=YMAX                    <NM>
2725 DPOKE $CB,X:POKE $CD,Y                   <WB>
2730 ELSE :X=DPEEK($CB):Y=PEEK($CD):STR=
STRIG(%0):ENDIF                               <NG>
2750 IF MODE=9 THEN I=8/APIX:X=X DIV I*I <HC>
2770 IF GRS=8 AND STR=1 THEN EXEC CRSGR8 <ZO>
2800 IF FKTFG AND MODE<>%1 AND STR AND X
=XA AND Y=YA:GO# HAUPT:ENDIF :XA=X:YA=Y  <JZ>
2950 ON MODE GO# DRAW,LINE,LINEX,KREIS,K
REIS,RECHT,RECHT,FILL,COPY,TEXTG,ZOOM    <HN>
2995 REM ▲▲ ZEICHNEN                     <JK>
3000 # DRAW:IF STR THEN FF=%0:GO# HAUPT  <TT>
3020 IF FF:DRAWTO X,Y                    <BI>
3040 ELSE :PLOT X,Y:FF=%1                <CD>
3090 ENDIF :GO# HAUPT                    <FQ>
3190 REM ▲▲ PUNKTE                       <BT>
3200 # PUNKT:IF STR=%0 THEN PLOT X,Y     <HM>
3290 GO# HAUPT                           <FC>
3495 REM ▲▲ LINIEN                       <PM>
3500 # LINE:IF FF AND STR THEN MOVE SC2,
SC1,SCL:PLOT X1,Y1:DRAWTO X,Y:GO# HAUPT  <DU>
3600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <IF>
3650 IF STR=%0 THEN FF=FF=%0:PLOT X1,Y1:
DRAWTO X,Y:EXEC BUT                      <NZ>
3690 GO# HAUPT                           <FK>
3699 REM ▲▲ LINIEN VERBUNDEN             <QK>
3700 # LINEX:IF PEEK(N764)=42 THEN FF=%0
:POKE N764,N255:MOVE SC2,SC1,SCL         <AH>
3710 IF FF AND STR THEN MOVE SC2,SC1,SCL
:PLOT X1,Y1:DRAWTO X,Y:GO# HAUPT         <MW>
3750 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:PLOT X,Y:X1=X:Y1=Y:FF=%1:EXEC BUT
:GO# HAUPT                               <RJ>
3800 IF STR=%0 THEN PLOT X1,Y1:DRAWTO X,
Y:X1=X:Y1=Y:MOVE SC1,SC2,SCL:EXEC BUT    <CX>
3990 GO# HAUPT                           <FQ>
3995 REM ▲▲ KREIS                        <WE>
4000 # KREIS:IF MODE=5:RD1=ABS(X-X1):RD2
=ABS(Y-Y1)                               <VR>
4020 ELSE :RD1=SQR((X-X1)^2+((Y-Y1)/(API
X/ASLI))^2):RD2=RD1*(APIX/ASLI):ENDIF    <LG>
4100 IF FF AND STR THEN MOVE SC2,SC1,SCL
:CIRCLE X1,Y1,RD1,RD2:GO# HAUPT          <OQ>
4200 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <HY>
4300 IF STR=%0 THEN FF=FF=%0:CIRCLE X1,Y
1,RD1,RD2:EXEC BUT                       <TM>
4490 GO# HAUPT                           <FH>
4495 REM ▲▲ RECHTECK                     <PQ>
4499 # RECHT                             <GI>
4500 IF FF AND STR THEN MOVE SC2,SC1,SCL
:PLOT X1,Y1:DRAWTO X1,Y:DRAWTO X,Y:DRAWT
O X,Y1:DRAWTO X1,Y1:GO# HAUPT            <NI>
4600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X1=X:Y1=Y:FF=%1:EXEC BUT:GO# HAUP
T                                        <IG>
4700 IF STR=%0:FF=FF=%0:PLOT X1,Y1:DRAWT
O X1,Y:DRAWTO X1,Y1:DRAWTO X,Y1:DRAWTO X
1,Y1                                     <DS>
4720 IF MODE=7 THEN N=1-(2*(Y1>Y)):FOR I
=Y1 TO Y STEP N:PLOT X1,I:DRAWTO X,I:NEX
T I                                      <MA>
4750 EXEC BUT:ENDIF                      <BF>
4900 GO# HAUPT                           <EQ>
4995 REM ▲▲ AUSFUELLEN                   <XW>
5000 # FILL:IF PEEK(N764)=28 THEN MOVE S
C2,SC1,SCL:POKE N764,N255                <XN>
5080 IF STR=%0:IF GRS=8 THEN X=X DIV %2  <AS>
5100 PAINT X,Y:ENDIF                     <BT>
5200 GO# HAUPT                           <ED>
5500 REM ▲▲ KOPIEREN                     <AX>
5520 # COPY                              <CN>
5550 IF CPF=%0                           <TK>
5600 IF FF=%0 AND STR=%0 THEN MOVE SC1,S
C2,SCL:X2=X:Y2=Y:FF=%1:GO# HAUPT         <IE>
5619 IF FF AND STR:MOVE SC2,SC1,SCL      <QM>
5620 PLOT X2,Y2:DRAWTO X2,Y:DRAWTO X,Y:D
RAWTO X,Y2:DRAWTO X2,Y2:GO# HAUPT:ENDIF  <EX>
5700 IF STR=%0:FF=%0:CPF=%1:X4=X:Y4=Y:X3
=X2:Y3=Y2:XD=ABS(X4-X3):YD=ABS(Y4-Y3):EX
EC BUT                                   <XA>
5710 IF XD<%1 OR YD<%1 THEN CPF=%0:FF=%1
:ENDIF                                   <CD>
5730 I=8/APIX:Z1=SC1:XDGER=INT(-XD/I)*-%
1:YDG=YD:MOVE SC2,SC1,SCL                <SF>
5750 ELSE                                <UB>
5760 IF X>XMAX-XD THEN X=XMAX-XD         <EQ>
5770 IF Y>YMAX-YD THEN Y=YMAX-YD         <JG>
5800 IF PEEK(N764)=63                    <LR>
5830 PLOT X3,Y3:DRAWTO X3,Y4:DRAWTO X4,Y
4:DRAWTO X4,Y3:DRAWTO X3,Y3              <YL>
5880 PLOT X,Y:DRAWTO X+XD,Y:DRAWTO X+XD,
Y+YD:DRAWTO X,Y+YD:DRAWTO X,Y            <SG>
5900 ELSE                                <TQ>
5910 Z=USR(DKAD+320,Z1-SC1+SC2,Z1,XDGER,
YDG,ABYT)                                <TV>
5920 I=8/APIX:XDGER=INT(-XD/I)*-%1       <JW>
5930 Q1=SC2+Y3*ABYT+X3 DIV I:Z1=SC1+Y*AB
YT+X DIV I:YDG=YD                        <NF>
5940 Z=USR(DKAD+320,Q1,Z1,XDGER,YDG,ABYT
)                                        <LB>
5980 IF STR=%0 THEN Z=USR(DKAD+320,Z1,Z1
-SC1+SC2,XDGER,YDG,ABYT):X3=X:Y3=Y:X4=X+
XD:Y4=Y+YD:ENDIF                         <AK>
5982 IF PEEK(N764)=42 THEN CPF=%0:MOVE S
C2,SC1,SCL:POKE N764,N255:ENDIF          <BO>
5990 GO# HAUPT                           <FS>
5999 REM ▲▲ TEXT                         <JV>
6000 # TEXTG:IF STR THEN FF=%1:MOVE SC2,
SC1,SCL:TEXT X,Y,TXT$:IF PEEK(N764)=45:G
OTO 6400:ENDIF :GO# HAUPT                <IO>
6300 IF STR=%0 THEN MOVE SC1,SC2,SCL:TEX
T X,Y,TXT$                               <IO>
6390 GO# HAUPT                           <FH>
6400 POKE N764,N255:TXT$="":N=%1         <VT>
6405 MOVE SC2,SC1,SCL                    <VO>
6410 DO :GET I:IF I=155:EXIT :ELSE :TXT$
(N,N)=CHR$(I):N=N+%1:MOVE SC2,SC1,SCL:TE
XT X,Y,TXT$:LOOP :ENDIF                  <ZK>
6480 POKE N764,N255                      <DZ>
6490 GO# HAUPT                           <FJ>
6499 REM ▲▲▲ ZOOM                        <RY>
6500 # ZOOM:ZMFG=%1                      <QO>
6520 CPF=1:XD=40/(APIX/ASLI)-%1:YD=23    <EG>
6550 IF FF=%0                            <HK>
6560 IF X>XMAX-XD THEN X=XMAX-XD         <EN>
6570 IF Y>YMAX-YD THEN Y=YMAX-YD         <JD>
6580 IF STR                              <LL>
6600 MOVE SC2,SC1,SCL                    <UY>
6630 PLOT X,Y:DRAWTO X+XD,Y:DRAWTO X+XD,
Y+YD:DRAWTO X,Y+YD:DRAWTO X,Y            <RO>
6680 ELSE :FF=%1:MOVE SC2,SC1,SCL        <PM>
6690 X1=X:Y1=Y:REM OGZ=GZ:GZ=1:GOSUB 800
0:GZ=OGZ                                 <KS>
6700 Z=USR(DKAD+400,%0,PUFAD,X,XD+%1,Y,Y
D+%1,APIX)                               <MQ>
6770 ENDIF :GO# HAUPT                    <GB>
6780 ENDIF                               <TX>
6800 I=X DIV 8:J=Y DIV 8:X=I*8:Y=J*8     <FW>
6820 COLOR PEEK(PUFAD+Y2*(XD+%1)+X2):TEX
T X2*8,Y2*8,"▲":COLOR COL:TEXT X,Y,"(CTL
 S)":X2=I:Y2=J                           <OF>
6850 IF STR=%0 THEN TEXT X,Y,"▲":POKE PU
FAD+J*(XD+%1)+I,COL                      <GH>
6900 IF PEEK(N764)=42:POKE N764,N255     <VM>
6910 MOVE SC2,SC1,SCL                    <VH>
```

Listing 1. »Grafik-Draw«

```
6920 Z=USR(DKAD+400,%1,PUFAD,X1,XD+%1,Y1
,YD+%1,APIX)                               <MX>
6940 MOVE SC1,SC2,SCL:FF=%0:X=X1:Y=Y1:DP
OKE $CB,X:POKE $CD,Y:GOSUB 8000:GO# ZOOM   <UO>
6995 ENDIF :GO# HAUPT                      <HF>
7000 REM ▲▲▲ MENU                          <HC>
7050 IF FF=%0 AND MODE<>11 AND MODE<>9 T
HEN MOVE SC1,SC2,SCL                       <YQ>
7100 POKE $060D,%0:POKE $060F,%0:POKE $0
618,%0                                     <NS>
7200 GRAPHICS %0:POKE 559,%0:POKE 53249,
%0:POKE 53250,%0:POKE 710,10:POKE 709,0:
POKE 712,80                                <XG>
7210 POKE N764,N255                        <DB>
7250 MPS=MODE:POKE $060C,191:POKE $CD,MO
DE*6:Y=MODE*8+56:GOSUB 9000                <FY>
7300 POKE 752,%1:? "{ESC CTL <}{CTL Q}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 E}"                                       <BB>
7350 ? "{SHIFT =}▲▲▲▲▲▲▲▲▲▲▲▲Grafik▲Draw
":POSITION 37,%1:? "{SHIFT =}"             <PU>
7380 ? "{SHIFT =}▲▲▲▲(c)▲1987▲▲Klaus▲Kra
ssowski▲":POSITION 37,%2:? "{SHIFT =}"     <BJ>
7450 ? "{CTL A}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL W}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL C}"                      <YQ>
7600 ? "{SHIFT =}▲Zeichnen▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <ZN>
7605 ? "{SHIFT =}▲Linie▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <JC>
7610 ? "{SHIFT =}▲Linie▲folgend▲▲▲{SHIFT
 =}"                                       <KY>
7615 ? "{SHIFT =}▲Kreis▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <QN>
7620 ? "{SHIFT =}▲Ellipse▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <EN>
7625 ? "{SHIFT =}▲Rechteck▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <UV>
7630 ? "{SHIFT =}▲Rechteck▲gef.▲▲▲{SHIFT
 =}"                                       <UD>
7640 ? "{SHIFT =}▲Ausfuellen▲▲▲▲▲▲{SHIFT
 =}"                                       <OZ>
7650 ? "{SHIFT =}▲Kopieren▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <FV>
7652 ? "{SHIFT =}▲Text▲▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <HW>
7654 ? "{SHIFT =}▲Zoom▲▲▲▲▲▲▲▲▲▲▲▲{SHIFT
 =}"                                       <HB>
7656 ? "{CTL A}▲Farben▲▲▲▲▲▲▲▲▲{CTL R}{C
TL S}{CTL R}{CTL R}{CTL R}{CTL R}{CTL E}
{ESC CTL =}{ESC CTL +}{SHIFT =}{ESC CTL
-}"                                        <FO>
7660 ? "{SHIFT =}▲Grafik▲Stufe▲▲▲▲:[";GR
S;"]"                                      <EG>
7670 ? "{SHIFT =}▲Loeschen▲▲▲▲▲▲▲▲{CTL A
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL X}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL E}{ESC CT
L =}{ESC CTL +}{SHIFT =}{ESC CTL -}"       <XP>
7680 ? "{SHIFT =}▲Disk▲▲▲▲▲▲▲▲▲▲▲▲:";GRN
$                                          <CH>
7690 ? "{SHIFT =}▲Drucken▲▲▲▲▲▲▲▲▲{CTL A
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL D}{ESC CT
L =}{ESC CTL +}{SHIFT =}{ESC CTL -}"       <SV>
7692 ? "{SHIFT =}▲Zeichengeraet▲▲▲:";      <GI>
7695 IF JTF:? "Touch▲Tablet":ELSE :? "Jo
ystick":ENDIF                              <KU>
7698 ? "{CTL Z}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{CTL R}{CTL X}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL C}":? :POKE 752,%0              <ST>
7700 DO : IF JTF=%0:Y=PEEK($CD):A=Y DIV 6  <WO>
7705 ELSE :Y=200-PADDLE(%3):Y=Y*(Y>%0):A
=Y DIV 10:PAUSE %1:ENDIF                   <QK>
7710 ST1=STRIG(%0):ST2=STICK(%1)-14        <XQ>
7714 IF A AND MPS<>A:MPS=A                 <FB>
7716 IF MPS>17:MPS=17:POKE $CD,MPS*6:ELS
E                                          <NW>
7718 SOUND %0,MPS*8+20,10,LAUTS:PAUSE %1   <JH>
7720 Y=MPS*8+56:GOSUB 9000                 <WW>
7725 ENDIF :SOUND %0,%0,%0,%0:ENDIF :IF
ST1=%0 OR ST2=%0 THEN EXIT                 <GC>
7730 LOOP                                  <YX>
7770 IF MPS<12 THEN MODE=MPS               <LB>
7800 IF MPS=13 THEN GOSUB 14000            <QX>
7810 IF GRS=5 AND (MODE=9 OR MODE=11 OR
MPS=16) THEN GOSUB 9800:MODE=1:GOTO 7100   <OV>
7820 IF MPS=14 THEN ? "Bild▲Loeschen▲[J/
N]▲";:GET I:IF I<>ASC("J") THEN 7300       <MC>
7830 IF MPS=10 THEN ? "Text▲->▲:";:INPUT
 TXT$                                      <YF>
7850 IF MPS=15 THEN GOSUB 17000            <VU>
7870 IF MPS=17 THEN JTF=JTF=%0:POKE $061
5,JTF:GOTO 7300                            <EV>
7880 IF MODE=9 THEN POKE $0618,%1          <RE>
7900 IF MPS=16 THEN GOSUB 21000            <QK>
7940 REM                                   <UI>
7950 POKE 53249,%0:POKE 53250,%0           <PT>
7960 POKE 559,%0:GRAPHICS GRS+16:SC1=DPE
EK(88)                                     <EA>
7962 GOSUB 8000:GOSUB 15000                <JA>
7965 POKE $060C,YMAX:MOVE SC2,SC1,SCL      <FA>
7970 IF GRS=8 THEN POKE 712,%2             <RJ>
7975 IF MPS=14 THEN CLS #6:MOVE SC1,SC2,
SCL                                        <JC>
7980 IF MPS=16 AND DRFG THEN GOSUB 20000
:GOTO 7000                                 <OR>
7985 IF MPS=12 THEN GOSUB 12500            <VO>
7990 IF MPS=15 THEN ON LSF GOSUB 18000,1
8500:GOSUB 15000:MOVE SC1,SC2,SCL:GOTO 7
000                                        <OJ>
7995 GOTO 2600                             <SM>
8000 REM ▲▲▲ PM STEUERUNG                  <EH>
8050 NR=%0:RESTORE 8100+GZ*10              <ER>
8060 DO :READ A:IF A=-1 THEN EXIT          <AL>
8080 NR=NR+%1:PM$(NR,NR)=CHR$(A)           <SA>
8090 LOOP :POKE $060E,NR                   <YT>
8110 DATA▲0,0,48,48,72,72,72,72,48,48,0,
-1                                         <FI>
8120 DATA▲0,0,32,32,80,80,32,32,0,-1       <WB>
8130 DATA▲0,0,48,48,204,48,48,0,-1         <OQ>
8140 DATA▲0,0,24,24,102,24,24,0,-1         <RS>
8150 DATA▲0,0,48,48,204,48,48,0,-1         <OW>
8160 DATA▲0,0,32,32,80,32,32,0,0,0,-1      <ZD>
8170 DATA▲0,0,0,0,0,-1                     <DI>
8200 POKE 559,62:POKE 705,15               <ON>
8210 I=PEEK(106)-40:PMAD=I*256+1280        <XA>
8220 POKE 54279,I:POKE 53277,%3            <QP>
8260 POKE 53257,%0:REM PLAYER 1 !!         <LB>
8262 POKE 53258,%1                         <EC>
8300 POKE PMAD,%0:MOVE PMAD,PMAD+%1,N255   <CR>
8310 POKE PMAD+N255,%0:MOVE PMAD+N255,PM
AD+256,N255                                <QG>
8320 POKE PMAD+N255+20,N255:MOVE PMAD+N2
55+20,PMAD+N255+21,10                      <LT>
8350 POKE 53250,48                         <NU>
8490 RETURN                                <QJ>
8990 REM ▲▲ PM MENU                        <FM>
9000 POKE PMAD,%0:MOVE PMAD,PMAD+%1,512    <JK>
9050 POKE 559,62:POKE 53277,%3:POKE 623,
%1                                         <PN>
9055 IF MPS<12:POKE 705,186:POKE 706,186
:ELSE :POKE 705,42:POKE 706,42:ENDIF       <NW>
9070 POKE 53257,%3:POKE 53258,%3:POKE 53
249,62:POKE 53250,90                       <WR>
9100 MOVE ADR(PM2$),PMAD+Y,8               <NN>
```

```
9120 MOVE ADR(PM2$),PMAD+256+Y,8                <TZ>
9190 RETURN                                      <QE>
9499 REM ▲ PM AUS                                <VU>
9500 POKE 710,16:POKE 709,12:POKE 705,16
:POKE 706,16:RETURN                              <HP>
9799 REM ▲ TON                                   <JO>
9800 SOUND %0,90,10,LAUTS:PAUSE 8:SOUND
%0,%0,%0,%0:RETURN                               <LR>
10000 REM ▲▲▲ INIT                               <GE>
10020 DIM PM$(30),PM2$(13),GRSII$(7700),
PUF$(1000),CPF$(320)                             <QJ>
10030 DIM A$(15),GRN$(15),TXT$(40)               <TO>
10040 DIM GCOL(7,8)                              <VH>
10050 SC2=ADR(GRSII$):PMSTR=ADR(PM$)             <YL>
10080 DIM GR$(6),KLR$(40)                        <VW>
10150 PM2$="{ESC CTL >}":PM2$(12)="{ESC
CTL >}":PM2$(%2)=PM2$                            <ON>
10190 GRSII$="{CTL ,}":GRSII$(1300)="{CT
L ,}":GRSII$(%2)=GRSII$:PUF$=GRSII$              <PE>
10200 KLR$="▲":KLR$(40)="▲":KLR$(%2)=KLR
$:GRN$=KLR$                                      <MQ>
10220 PUFAD=ADR(PUF$)                            <HR>
10230 GR$=CHR$(27):GR$(%2)="K"                   <VK>
10330 N255=255:N764=764                          <SV>
10350 COL=%1:ACOL=COL:DFMT=%1:DRFG=%0:JT
F=%0:JTMP=%0:INVFG=%0:LAUTS=4:TEIL=%2            <XH>
10400 MODE=%1                                    <QC>
10410 GZ=6:GOSUB 14100:GOSUB 14500               <RV>
10490 RETURN                                     <UT>
12000 REM ▲▲ FARB WAHL                           <AC>
12020 GOSUB 9800:A=CM:IF GRS=9 OR GRS=11
 THEN A=16                                       <ZD>
12030 IF GRS=8 AND MODE=8 THEN POKE 87,1
5:A=4:X=X DIV %2                                 <MV>
12080 ACOL=COL                                   <IJ>
12100 COL=COL+%1-(COL+%1)*(COL>=A-%1)            <CV>
12120 COLOR COL                                  <SC>
12200 IF GRS>8 AND GRS<12 OR (GRS=8 AND
MODE=8):POKE 53250,%0:MOVE SC1,ADR(CPF$)
,320:TEXT %0,%0,"▲▲"                             <XE>
12220 PAUSE 30:MOVE ADR(CPF$),SC1,320:EL
SE :POKE 706,GCOL(CMEN*%2+%1,COL)                <XV>
12300 ENDIF :POKE N764,N255                      <BG>
12390 RETURN                                     <UU>
12490 REM ▲▲ FARB MENU                           <KE>
12500 A=CM:A$="▲▲▲▲▲▲▲▲":IF GRS=9 OR GRS
=11 THEN A=16                                    <YG>
12600 FOR I=%0 TO A-%1                           <AQ>
12640 COLOR I:TEXT %0,I*12,A$(%1,8/APIX)         <YD>
12660 NEXT I:COL=%0:TEXT 64/APIX,%0,"{ES
C CTL +}"                                        <FX>
12700 DO :A=PEEK(53279):J=CMEN*%2                <OP>
12720 IF A=6 THEN TEXT 64/APIX,COL*12,"▲
":COL=(COL+%1)*(COL<CM-%1):TEXT 64/APIX,
COL*12,"{ESC CTL +}":GOSUB 9800                  <LH>
12740 I=GCOL(J+%1,COL):H=I MOD 16:F=I-H          <HB>
12760 IF A=%3 THEN F=(F+16)*(F<240):GCOL
(J+%1,COL)=F+H:POKE GCOL(J,COL),F+H:PAUS
E 10                                             <JC>
12800 IF A=5 THEN H=(H+%2)*(H<14):GCOL(J
+%1,COL)=F+H:POKE GCOL(J,COL),F+H:PAUSE
10                                               <AC>
12880 IF PEEK(N764)=28 THEN POKE N764,N2
55:EXIT                                          <LO>
12900 LOOP                                       <WT>
12990 MOVE SC2,SC1,SCL:RETURN                    <BU>
14000 REM ▲▲ GRAPHIC WAHL                        <UI>
14020 ACOL=%0:GOSUB 9500:? "{ESC CTL <}{
ESC CTL =}"                                      <RU>
14050 ? "Welche▲Grafik-Stufe?":? :? "1▲=
▲[▲5]":? "2▲=▲[▲7]":? "3▲=▲[▲9]":? "4▲=▲
[10]":? "5▲=▲[11]":? "6▲=▲[15]"                  <JL>
14060 ? "7▲=▲[▲8]"                               <ZH>
14070 REPEAT :GET I:I=I-48:UNTIL I>%0 AN
D I<8                                            <IL>
14080 GZ=I                                       <CR>
14100 RESTORE 14390+GZ*10:READ GRS,SCL,X
MAX,YMAX,ABYT,APIX,ASLI,CMEN,CM                  <IS>
14210 DPOKE $060A,XMAX:POKE $060C,YMAX           <EG>
14220 POKE $0611,APIX:POKE $0612,ASLI            <AV>
14290 RETURN                                     <UV>
14392 REM GRS,SCL,XMAX,YMAX,ABYT,APIX,AS
LI                                               <NV>
14400 DATA▲5,960,79,47,20,4,4,1,4                <IN>
14410 DATA▲7,3840,159,95,40,2,2,1,4              <HT>
14420 DATA▲9,7680,79,191,40,4,1,3,1              <JI>
14430 DATA▲10,7680,79,191,40,4,1,2,9             <FD>
14440 DATA▲11,7680,79,191,40,4,1,3,1             <WA>
14450 DATA▲15,7680,159,191,40,2,1,1,4            <NL>
14460 DATA▲8,7680,319,191,40,1,1,0,2             <QD>
14490 REM ▲▲ COLORS                              <CP>
14500 RESTORE 14600                              <LG>
14510 FOR N=%0 TO %3:READ J                      <PG>
14520 FOR I=%0 TO J-%1                           <HO>
14530 READ A,B:GCOL(N*%2,I)=700+A                <FW>
14540 GCOL(N*%2+%1,I)=B                          <ZQ>
14550 NEXT I:NEXT N                              <UB>
14590 RETURN                                     <VE>
14600 DATA▲2,10,0,9,15                           <WB>
14610 DATA▲4,12,0,8,186,9,150,10,40              <DN>
14620 DATA▲9,4,8,5,14,6,184,7,150,8,32,9
,176,10,90,11,80,12,60                           <EQ>
14630 DATA▲1,12,8                                <UB>
15000 REM ▲▲ FARBEN SETZEN                       <MR>
15010 FOR I=%0 TO CM-%1:J=CMEN*%2                <CH>
15050 POKE GCOL(J,I),GCOL(J+%1,I)                <XJ>
15100 NEXT I:RETURN                              <AM>
17000 REM ▲▲ FILENAME                            <NQ>
17005 TRAP 7200:GOSUB 9500:? "{ESC CTL <
}{ESC CTL =}Disk▲Menu"                           <HW>
17010 DO :LSF=%0:?                               <PY>
17020 ? "[D]▲Directory"                          <MA>
17030 ? "[L]▲Laden"                              <IK>
17040 ? "[S]▲Speichern"                          <ZQ>
17050 ? "[{ESC ESC}{ESC ESC}]▲ENDE":?            <FC>
17060 GET I                                      <TF>
17100 IF I=76 THEN LSF=%1:EXIT                   <RS>
17110 IF I=83 THEN LSF=%2:EXIT                   <RU>
17120 IF I=68 THEN ? :? :DIR                     <XX>
17125 IF I=27 THEN EXIT                          <OB>
17130 LOOP :IF I=27 THEN 17300                   <EW>
17140 PUT I+128:? :? :IF LSF=%2 THEN ? ,
"{ESC CTL *}";GRN$;"{ESC CTL -}"                 <NR>
17150 INPUT "Filename▲{ESC ESC}{ESC CTL
*}";A$                                           <OR>
17160 A=INSTR(A$,":"):IF A=%2 OR A=%3 TH
EN GRN$=A$:RETURN                                <DI>
17180 GRN$(%1)="D:":GRN$(%3)=A$                  <MM>
17300 RETURN                                     <TU>
18000 REM ▲▲▲ LADEN                              <RW>
18100 TRAP 18490:OPEN #4,4,0,GRN$                <EX>
18150 ACOL=%0:CLS #6:BGET #4,SC1,SCL             <DO>
18200 GET #4,I:IF I=75                           <HR>
18250 GET #4,GZ:GOSUB 14100                      <QZ>
18300 GET #4,CM                                  <ZD>
18320 FOR I=%0 TO CM-%1:J=CMEN*%2                <DA>
18340 GET #4,A:GET #4,B                          <KP>
18350 GCOL(J,I)=A+700:GCOL(J+%1,I)=B             <IV>
18400 NEXT I                                     <GL>
18490 ENDIF :CLOSE #4:RETURN                     <RZ>
18500 REM ▲▲▲ SPEICHERN                          <QP>
18600 TRAP 18900:OPEN #4,8,0,GRN$                <GZ>
18700 BPUT #4,SC1,SCL                            <ZG>
18800 PUT #4,75:PUT #4,GZ:PUT #4,CM              <CY>
18820 FOR I=%0 TO CM-%1:J=CMEN*%2                <DP>
18830 A=GCOL(J,I)-700:B=GCOL(J+%1,I)             <ET>
18850 PUT #4,A:PUT #4,B                          <IR>
18860 NEXT I                                     <HV>
18900 CLOSE #4:RETURN                            <FK>
20000 REM ▲▲▲ DRUCK ROUTINE                      <OY>
20100 TRAP 20600:OPEN #3,8,0,"P:"                <FN>
20120 IF INVFG THEN Z=USR(DKAD+280,YMAX+
%1)                                              <GY>
20150 ? #3;"{ESC ESC}A{CTL H}"                   <BH>
```

Listing 1. »Grafik-Draw« (Fortsetzung)

```
20200 FOR N=SC1 TO SC1+SCL-320 STEP 320  <FB>
20250 IF PEEK(N764)=126 THEN POKE N764,N
255:EXIT                                  <KP>
20300 Z=USR(DKAD,DFMT,N,SC2)              <CO>
20350 IF DFMT=%1 THEN ? #3;KLR$(%1,DPS);
GR$;GRSII$(%1,320)                        <RU>
20400 IF DFMT=%2 THEN ? #3;GR$;GRSII$(%1
,640):? #3;GR$;GRSII$(641,1280)           <CN>
20450 NEXT N                              <IY>
20500 IF INVFG THEN Z=USR(DKAD+280,YMAX+
%1)                                       <HC>
20550 MOVE SC1,SC2,SCL                    <UX>
20600 CLOSE #3:RETURN                     <DY>
21000 REM ▲▲▲ DRUCK FORMAT                <AW>
21050 GOSUB 9500                          <LV>
21100 REPEAT :? "{ESC CTL <}{ESC CTL =}{
ESC CTL =}DRUCK▲Format":? "{ESC CTL =}[1
]▲Halbe▲DIN▲A4▲Breite":? "[2]▲Ganze▲DIN▲
A4▲Breite":?                              <GH>
21120 GET A:PUT A+128:A=A-48              <SB>
21150 UNTIL A>%0 AND A<%3                 <UF>
21200 DFMT=A                              <IC>
21250 IF DFMT=%1 THEN GR$(3,3)=CHR$(64):
GR$(4,4)=CHR$(1)                          <MO>
21450 IF DFMT=%2 THEN GR$(3,3)=CHR$(128)
:GR$(4,4)=CHR$(2):GOTO 23000              <UC>
21750 REPEAT :? "{ESC CTL <}{ESC CTL =}{
ESC CTL =}DRUCK▲Position":? "{ESC CTL =}
[1]▲Links":? "[2]▲Mitte":? "[3]▲Rechts":
? "[4]▲Individuel"                        <LV>
21800 ? :GET A:PUT A+128:A=A-48:UNTIL A>
%0 AND A<5                                <GN>
21850 IF A<>4:DPS=(A-%1)*19+%1            <VE>
22150 ELSE :? :REPEAT :? "{ESC CTL -}{ES
C SHIFT DEL}Position▲[1-40]▲:";:INPUT DP
S:UNTIL DPS>%0 AND DPS<41                 <TP>
23000 ENDIF                               <RZ>
23020 ? :? "{ESC CTL =}Invers▲Drucken▲▲[
J/N]▲";:GET I:IF I=ASC("J"):INVFG=%1:ELS
E :INVFG=%0:ENDIF                         <ZP>
23500 ? :? "{ESC CTL =}Starte▲Drucken▲▲[
J/N]▲";:GET I                             <DL>
23510 IF I=ASC("J"):DRFG=%1:ELSE :DRFG=%
0:INVFG=%0:ENDIF                          <WE>
24000 RETURN                              <TG>
```

Listing 1. »Grafik-Draw« (Schluß)

```
Programmname :UPADSTK.BIN
Länge        :400 Bytes
-----------------------------------

0000:FF FF 00 98 86 99 68 68 <0A>
0008:85 CF 68 85 CE 68 68 8D <5D>
0010:0E 06 A9 00 8D 0F 06 A9 <03>
0018:14 8D 01 06 A2 98 A0 20 <90>
0020:A9 07 20 5C E4 60 AD 15 <83>
0028:06 F0 55 AD 72 02 18 6D <DD>
0030:02 06 8D 02 06 90 03 EE <CA>
0038:03 06 AD 73 02 18 6D 04 <57>
0040:06 8D 04 06 90 03 EE 05 <F2>
0048:06 EE 00 06 AD 00 06 CD <65>
0050:01 06 D0 2C AD 02 06 8D <15>
0058:06 06 AD 03 06 8D 07 06 <19>
0060:AD 04 06 8D 08 06 AD 05 <F7>
0068:06 8D 09 06 A9 00 8D 00 <66>
0070:06 8D 02 06 8D 03 06 8D <40>
0078:04 06 8D 05 06 4C BB 98 <4C>
0080:AE 0D 06 E0 02 B0 6C AE <2F>
0088:78 02 8A 29 04 D0 15 A5 <57>
0090:CC D0 04 A5 CB F0 0D 38 <21>
0098:A5 CB E9 01 85 CB A5 CC <B9>
00A0:E9 00 85 CC 8A 29 08 D0 <3B>
00A8:06 E6 CB D0 02 E6 CC 8A <64>
00B0:29 01 D0 06 A5 CD F0 02 <3A>
00B8:C6 CD 8A 29 02 D0 02 E6 <CB>
00C0:CD A5 CC CD 0B 06 90 18 <3E>
00C8:F0 03 4C D0 98 A5 CB CD <C5>
00D0:0A 06 90 0C F0 0A AD 0A <15>
00D8:06 85 CB AD 0B 06 85 CC <A6>
00E0:AE 0C 06 E4 CD B0 02 86 <6C>
00E8:CD AE 0D 06 F0 05 A2 02 <D7>
00F0:8E 0D 06 AE 0F 06 D0 03 <C9>
00F8:4C 62 E4 18 AD 10 06 69 <4F>
0100:1C A8 A2 00 8A 99 00 9D <BA>
0108:C8 E8 EC 0E 06 D0 F6 A5 <C7>
0110:CD 8D 16 06 AE 12 06 E0 <E1>
0118:02 D0 07 18 2E 16 06 4C <83>
0120:27 99 E0 04 D0 07 18 2E <18>
0128:16 06 2E 16 06 18 AD 16 <DF>
0130:06 8D 10 06 69 1C AA A0 <6B>
0138:00 B1 CE 9D 00 9D E8 C8 <04>
0140:CC 0E 06 D0 F4 A5 CB 8D <C2>
0148:13 06 A5 CC 8D 14 06 AE <15>
0150:11 06 E0 02 F0 15 E0 01 <11>
0158:D0 0A 18 6E 14 06 6E 13 <55>
0160:06 4C 65 99 18 2E 13 06 <40>
0168:4C 65 99 AD 18 06 F0 10 <28>
0170:EE 13 06 18 4E 13 06 4E <79>
0178:13 06 0E 13 06 0E 13 06 <84>
0180:18 AD 13 06 69 2E 8D 01 <D4>
0188:D0 4C 62 E4 00 00 00 00 <23>
01E0:60 AD 84 06 9D 42 03 A9 <67>
01E8:00 9D 48 03 9D 49 03 A5 <00>
01F0:E2 F0 04 A0 00 B1 E0 20 <EC>
01F8:56 E4 A4 E2 D0 02 91 E0 <C0>
0200:E6 E0 D0 02 E6 E1 C6 D7 <A0>
0208:D0 D1 AD 82 06 85 D7 E6 <4C>
0210:D6 CE 83 06 D0 BD A5 E2 <40>
0218:F0 01 60 AD 86 06 85 E0 <50>
0220:AD 87 06 85 E1 AD 82 06 <6C>
0228:85 D7 A5 E3 8D 83 06 A5 <A5>
0230:E5 38 E9 01 0A AA BD 49 <9B>
0238:97 8D 6D 97 BD 4A 97 8D <CF>
0240:6E 97 A5 58 8D 80 06 A5 <76>
0248:59 8D 81 06 4C 51 97 6F <CE>
0250:97 8D 97 00 00 78 97 AD <71>
0258:80 06 85 D4 AD 81 06 85 <67>
0260:D5 18 A5 D4 69 A0 85 D8 <4B>
0268:A5 D5 69 00 85 D9 A0 00 <D1>
0270:B1 E0 4C 6F 97 F0 02 A9 <56>
0278:FF 85 D6 4C A0 97 85 E2 <E3>
0280:85 D6 06 E2 06 E2 06 E2 <85>
0288:06 E2 A5 E2 05 D6 85 D6 <3B>
0290:4C A0 97 85 E2 85 D6 A0 <5E>
0298:03 06 E2 06 E2 A5 E2 05 <A6>
02A0:D6 85 D6 88 D0 F3 A6 E5 <A0>
02A8:A0 00 91 D4 91 D8 A0 28 <99>
02B0:91 D4 91 D8 A0 50 91 D4 <3B>
02B8:91 D8 A0 78 91 D4 91 D8 <75>
02C0:E6 D4 D0 02 E6 D5 E6 D8 <ED>
02C8:D0 02 E6 D9 CA D0 D9 E6 <B4>
02D0:E0 D0 02 E6 E1 C6 D7 D0 <33>
02D8:95 AD 82 06 85 D7 18 AD <47>
02E0:80 06 69 40 8D 80 06 AD <72>
02E8:81 06 69 01 8D 81 06 CE <30>
02F0:83 06 F0 03 4C 51 97 60 <12>
02F8:00 00 00 00 00 00 00 00 <FC>
0300:00 00 00 00 00 00 00 00 <06>
```

Listing 2. Bitte mit AMPEL eingeben

```
Programmname :UKIPDRK.BIN
Länge        :776 Bytes
-----------------------------------

0000:FF FF 00 95 FF 97 68 68 <96>
0008:68 8D 83 06 68 85 E1 85 <9C>
0010:D8 68 85 E0 85 D7 68 85 <BA>
0018:E3 68 85 E2 A9 08 8D 82 <7A>
0020:06 AD 83 06 C9 02 D0 04 <11>
0028:18 6E 82 06 18 A5 E2 69 <B4>
0030:80 85 E4 A6 E3 90 01 E8 <AF>
0038:E8 E8 86 E5 A2 80 86 D4 <31>
0040:86 D5 A2 00 86 D9 8E 80 <D9>
0048:06 86 D6 8E 81 06 A0 00 <E5>
0050:B1 E0 25 D4 F0 1A 18 A5 <3F>
0058:D5 65 D6 85 D6 AD 83 06 <9D>
0060:C9 02 D0 16 18 66 D5 A5 <59>
0068:D5 65 D6 85 D6 18 90 0A <79>
0070:AD 83 06 C9 02 D0 03 18 <62>
0078:66 D5 66 D5 18 A5 E0 69 <F6>
0080:28 85 E0 90 02 E6 E1 E8 <4C>
0088:EC 82 06 D0 C1 A2 02 AC <49>
0090:81 06 C0 01 F0 12 A0 00 <E3>
0098:A5 D6 91 E2 E6 E2 D0 02 <8D>
00A0:E6 E3 18 90 0F 18 90 A6 <E6>
00A8:A0 00 A5 D6 91 E4 E6 E4 <B6>
00B0:D0 02 E6 E5 AD 83 06 C9 <6C>
00B8:02 D0 15 CA D0 D1 A9 80 <90>
00C0:85 D5 A2 00 86 D6 EE 81 <8D>
00C8:06 AC 81 06 C0 02 D0 D5 <E2>
00D0:18 66 D4 A5 D7 85 E0 A5 <7A>
00D8:D8 85 E1 A9 80 85 D5 A2 <3A>
00E0:00 86 D6 8E 81 06 E6 D9 <B8>
00E8:A4 D9 C0 08 D0 B7 85 D4 <A9>
00F0:86 D9 18 E6 D7 E6 E0 D0 <29>
00F8:04 E6 D8 E6 E1 EE 80 06 <93>
0100:AC 80 06 C0 28 D0 9E 60 <50>
0108:00 00 00 00 00 00 00 00 <0A>
0110:00 00 00 00 00 00 00 00 <12>
0118:00 00 00 00 00 00 68 68 <52>
0120:68 85 D6 A5 58 85 E0 A5 <06>
0128:59 85 E1 A0 00 A2 28 B1 <E6>
0130:E0 49 FF 91 E0 E6 E0 D0 <95>
0138:02 E6 E1 CA D0 F1 C6 D6 <21>
0140:D0 EB 60 00 00 00 68 68 <29>
0148:85 E1 68 85 E0 68 85 E3 <B8>
0150:68 85 E2 68 68 85 D4 68 <93>
0158:68 85 D5 68 68 85 D6 A6 <27>
0160:D5 A4 D4 88 B1 E0 91 E2 <0B>
0168:88 10 F9 18 A5 E0 65 D6 <1E>
0170:85 E0 90 02 E6 E1 18 A5 <66>
0178:E2 65 D6 85 E2 90 02 E6 <DE>
0180:E3 CA D0 DD 60 00 00 00 <B9>
0188:00 00 00 00 00 00 00 00 <8A>
0190:00 00 00 00 00 00 68 68 <CB>
0198:68 85 E2 68 85 E1 8D 87 <34>
01A0:06 68 85 E0 8D 86 06 68 <4C>
01A8:85 D5 68 85 D4 68 68 8D <92>
01B0:82 06 85 D7 68 68 85 D6 <35>
01B8:68 68 8D 83 06 85 E3 68 <12>
01C0:68 85 E5 A5 E2 D0 07 A9 <14>
01C8:07 8D 84 06 10 05 A9 0B <73>
01D0:8D 84 06 A5 D4 85 55 A5 <43>
01D8:D5 85 56 A5 D6 85 54 A2 <D7>
```

Listing 3. Bitte mit AMPEL eingeben

3D-Grafik in Turbo Basic

Dreidimensionale Computergrafiken sind immer häufiger Bestandteil von Film und Fernsehen. Mit dem Atari lassen sich bereits ansehnliche Grafiken berechnen und darstellen.

Nicht nur in der Industrie, in der Entwicklung oder im CAD-Bereich, sondern auch in Film und Fernsehen, besonders in der Werbung, bedient man sich immer mehr der »Bilder aus dem Computer«. Das ist kein Wunder, denn die Kosten für computeranimierte Filme sind drastisch gesunken. Computer, die solche Leistungen bringen, sind meistens teure Grafik-Workstations oder Supercomputer wie die »Cray« und besitzen neben schnellen Programmen spezielle Hardware. Dadurch wird die Grafik wesentlich schneller und viele Grafik-Routinen überflüssig. Eins steht auf jeden Fall fest: Die Berechnung von dreidimensionalen Grafiken mit verdeckten Kanten, Schattierung und Lichtreflexen braucht sehr viel Speicher und eine Menge Zeit.

Mit diesem Beitrag wollen wir nicht die mathematischen Grundlagen der dreidimensionalen Grafik erklären. Dafür behandeln wir das sogenannte »Clipping« oder »Windowing«, verdeckte Kanten und die Berechnung von Schlagschatten einfacher Objekte. Da bestimmte Unterprogramme immer wieder Verwendung finden, werden die einzelnen Prozeduren gesondert in einem Satz erläutert.

Steigen wir mit dem »Clipping« ein. Dahinter verbirgt sich nichts weiter als das Kappen von Linien, die über den Bildschirmrand hinausgehen. Ganz so einfach ist die Sache leider nicht. Ein wichtiger Faktor ist nämlich der Winkel, unter dem die Linie den Bildrand schneidet. Die Clipping-Routine berechnet nun im Prinzip die Länge und den Winkel des nicht sichtbaren Teils einer Linie gegenüber dem Bildrand und liefert als Ergebnis die erste wieder sichtbare Koordinate dieser Linie.

Der hier verwendete Clipping-Algorithmus wurde in Amerika von »Evans und Sutherland«, Pionieren der Computergrafik, entwickelt. Dieses Unternehmen liefert Computersimulationen hauptsächlich für militärische Zwecke.

Doch nun zum Programm: Innerhalb der Routine wird mit Hilfe der Prozedur »CODE« zunächst ein 4-Bit-Code berechnet, wobei die Bits folgende Bedeutung haben:

Bit 1: Punkt ist links vom Bildrand
Bit 2: Punkt ist rechts vom Bildrand
Bit 3: Punkt ist unterhalb vom Bildrand
Bit 4: Punkt ist oberhalb vom Bildrand

Die Variablen, die diese Bitkombinationen enthalten, sind C, C1 und C2. CZ dient hier nur als Tauschvariable. Die Bits sind entsprechend den Bitpositionen innerhalb eines Byte in den Variablen enthalten. Die Prozedur »CLIP« berechnet dann Länge und Winkel der verdeckten Linie(n). Dieser Prozeß wird solange durchlaufen, bis die Prozedur erkennt, daß die ganze Linie unsichtbar ist (Bedingung: VIS=0). Wenn Teile der Linie oder die ganze Linie sichtbar sind, so lautet die Bedingung VIS=1. Haben alle vier Bits den Wert 0, befindet sich der zu setzende Punkt oder die Linie innerhalb des Bildes und kann gesetzt werden. Die Variablen, die die Bildschirmgrenzen festsetzen, heißen:

CLIPYB	für den unteren Rand
CLIPYT	für den oberen Rand
CLIPXL	für den linken Rand
CLIPXR	für den rechten Rand

Durch Verändern dieser Variablen (im Listing Zeile 40 und 50) wird jeder beliebige Bildschirmbereich bestimmt. Das Objekt oder sichtbare Teile davon werden nur innerhalb dieses Fensters dargestellt. Obwohl wir hier mit dreidimensionalen Objekten arbeiten, ist die Clipping-Routine im Prinzip für zweidimensionale Objekte gedacht. Da auf dem Bildschirm aber zweidimensional abgebildet wird, kann die Routine verwendet werden. Es existiert zwar auch ein Cohen-Sutherland-Algorithmus, der dreidimensional arbeitet. Er ist aber nur von Interesse, wenn man sich in das Objekt »hinein«begeben will. Damit dieses dennoch mit der hier verwendeten Routine bedingt möglich ist, wird in der Prozedur »CALCPOINT« die Z-Koordinate eines Punktes automatisch auf 1 gesetzt, wenn diese vorher kleiner war. Damit wird verhindert, daß eventuell »vor« dem Bildschirm liegende Objekte durch falsche Berechnung ein Linienchaos auf dem Schirm erzeugen.

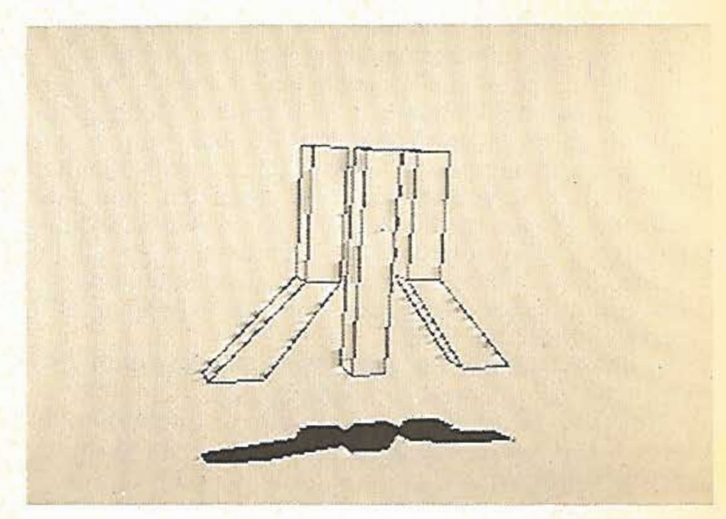


Bild 1. Ein bekanntes Zeichen in 3D

Ein nächstes Problem bereitet die Berechnung der verdeckten Kanten und Flächen (Bild 1). Über das »Hidden Line« oder »Hidden Surface«-Problem gibt es jede Menge Literatur, sogar Dissertationen und andere wissenschaftliche Arbeiten. Entsprechend groß ist auch die Anzahl der Methoden und Algorithmen zur Bewältigung dieses Problems. Die meisten Verfahren sind Scan-Algorithmen, bei denen das Bild Zeile für Zeile abgetastet, untersucht und dann entsprechend bearbeitet wird. Unsere Lösung ist ein Prioritäts-Algorithmus, der einen vertretbaren Kompromiß zwischen Aufwand und Genauigkeit darstellt. Es ist ein vereinfachter »Newell-Sancha«-Algorithmus. Zur weiteren Erklärung betrachten wir zuerst die Koordinaten-Verhältnisse.

Da wir hier dreidimensionale Objekte darstellen, braucht unser Koordinatensystem die Achsen X, Y und Z, wobei die Z-Koordinate senkrecht auf dem Bildschirm steht. Bei größer werdenden Z-Koordinaten bewegt man sich in den Bildschirm hinein.

Um verdeckte Kanten darzustellen, ist es von großer Bedeutung, wie das Objekt im Speicher des Computers abgelegt wird. Da wir von jetzt an nicht mehr nur Eckpunkte, sondern ganze Kanten des Objektes betrachten, müssen diese dem Programm als solche erkennbar sein. Zusätzlich zu diesen Informationen braucht der Computer noch Angaben über die Anzahl der Kanten jedes einzelnen Polygons und über die Verbindungslinien.

Ein Würfel besteht jetzt nicht mehr aus 8 Eckpunkten, sondern aus 6 Polygonen. Diese exakte Beschreibung des

Objektes ist eine grundlegende Bedingung, um eine Ansicht mit verdeckten Kanten zu berechnen.

Der Algorithmus funktioniert folgendermaßen: Nachdem die Rotationen und alle anderen Transformationen beendet sind, sortiert die Prozedur »SORTPOLY« alle Polygone nach der Z-Koordinate. Dadurch bekommen die Polygone eine Priorität zugewiesen. Das Polygon, das am weitesten vom Betrachter entfernt ist, erhält die niedrigste Priorität. Die Polygone werden dann in der Reihenfolge ihrer Priorität auf den Bildschirm gebracht. Zuerst wird das Polygon der niedrigsten, zuletzt das Polygon der höchsten Priorität gezeichnet. Dabei ist zu beachten, daß nicht nur die äußere Umrandung des Polygons (Linien, die das Polygon bilden) gezeichnet wird, sondern das ganze Polygon in Form einer Fläche. Im Programm erledigt das die Prozedur »DDA«. Und schon ist die 3D-Darstellung mit verdeckten Kanten fertig.

Die Sache hat aber leider einen Schönheitsfehler. Der Algorithmus funktioniert nämlich nicht immer. Es wird besonders dann kritisch, wenn sich Polygone in der Z-Ebene überlappen. Betrachten wir dazu einmal Bild 2. Wenn man nach Z-Koordinaten sortiert, erhält Polygon 2 eine höhere Priorität als Polygon 1, weil dessen Z-Wert größer ist als der von Polygon 2. Im Ergebnis würde zuerst Polygon 1, dann Polygon 2 gezeichnet werden. Von der Darstellung her ist das falsch. Dieser Fehler ließe sich durch Erweitern der Sortierkriterien zwar umgehen, was aber wieder auf Kosten der Rechenzeit gehen würde. Möchte man dennoch die Berechnungen korrekt durchführen, so ergeben sich mit Bezug auf Bild 2 folgende Sachverhalte:

Polygon 1 verdeckt Polygon 2 (im folgenden P1 und P2 genannt) nicht in dem Fall,
- daß ein einleitender Test der Z-Werte ergibt, daß sich P1 und P2 in der Z-Ebene nicht überlappen.
- daß ein Test der größten und kleinsten X- und Y-Werte der einzelnen Polygone ergibt, daß sich diese in XY-Richtung nicht überlappen.
- daß alle Kanten von P1 weiter vom Betrachter entfernt sind als die Ebene von P2. Hierzu werden die X- und Y-Werte von P1 in die Ebenengleichung (AX+BY+CZ+D=0) von P2 eingesetzt und die »Tiefe« von P2 berechnet.
- daß alle Kanten von P2 näher am Betrachter sind als die Ebene, die P1 aufspannt.
- daß ein kompletter Überlappungstest zeigt, daß sich die Polygone in X/Y-Richtung nicht überlappen.

Werden alle diese Tests durchlaufen, muß man eine Menge Geduld haben. Versuche mit vielen Objekten haben ergeben, daß man trotz ausschließlichen Sortieren nach Z-Werten fast immer die korrekte Darstellung erhält.

Die Berechnung von Schlagschatten ist gegenüber den verdeckten Kanten und Flächen eigentlich recht simpel. Grundkenntnisse der Vektorrechnung und analytischen Geometrie erleichtern das Verständnis. An der Mathematik kommt man (leider) nicht vorbei. Dazu betrachten wir nun Bild 3.

Es sei l der Vektor vom Koordinatenursprung zur punktförmigen Lichtquelle mit den Koordinaten (LX,LY,LZ) und p der Vektor zu einem beliebigen Punkt P mit den Koordinaten (PX,PY,PZ) einer beliebigen Kante eines Polygons. Dann gilt:

1) r=p-l (Vektorsubtraktion)

Wir erhalten einen neuen Vektor r mit den Koordinaten (RX,RY,RZ). Der Schlagschatten des Punktes P soll nun auf die Y-Ebene mit Y=0 fallen. Der sich ergebende Punkt soll PS heißen. Wir erhalten somit:

2) PS=l+S*r

Der Vektor vom Koordinatenursprung zum Punkt PS ergibt sich also aus dem Vektor l plus den verlängerten Vektor r. Hierzu wird der Vektor r mit einem Skalar S multipliziert und dann zum Vektor l geometrisch addiert. Da wir wissen, daß die Y-Koordinate des Punktes PS immer gleich 0 sein muß, ergibt sich für S eine lineare Gleichung in Y:

3) 0=LY+S*RY oder LY+S*RY=0 => S=-LY/RY

Bildlich gesehen ist der Schlagschatten eines Polygons eine Projektion dieses Polygons auf eine Ebene, in diesem Fall die Ebene Y=0.

Wem das alles zu theoretisch ist, der betrachte Bild 3. Dann sollte einiges klarer erscheinen.

Als letztes kann man sich überlegen, daß der Schlagschatten eines Objektes gleich der Summe der Schlagschatten seiner Polygone ist. Zeichnen wir also zunächst jedes Polygon und dann den Schlagschatten des Polygons, so erhalten wir als Ergebnis das Objekt und den Schlagschatten des Objekts.

Das abgedruckte Programm (Listing 1) ist als solches zwar in sich abgeschlossen und lauffähig, es soll aber auch eine Basis für weitere Experimente darstellen. Deshalb wurde bewußt auf aufwendige Menüs und »Kosmetik« verzichtet, weil das alles schließlich wertvollen Speicherplatz kostet. Des weiteren ist das ganze Programm in Basic geschrieben, so daß eine Adaption auf andere Computer keine großen Schwierigkeiten bereiten sollte.

Zunächst zum Hauptprogramm. Die wichtigsten Variablen und Felder werden gleich zu Beginn des Programms in den Zeilen 80 bis 120 festgelegt.
- Die Felder POLY_Z(100) und PRIO(100) enthalten Priorität und den größten Z-Wert der einzelnen Polygone, in diesem Fall für maximal 100 Einzelpolygone.
- Die Felder SX, SY, SZ, X, Y enthalten die Koordinaten der Eckpunkte der Schatten und die transformierten Koordinaten der Objekte oder Schattenpolygone. In diesem Fall darf jedes Polygon maximal 10 Eckpunkte besitzen.
- In den Zeilen 100 und 110 befinden sich die Variablen für die Bildschirmbegrenzungen. Es sind:

CLIPXL	kleinste X-Koordinate (linker Bildrand)
CLIPXR	größte X-Koordinate (rechter Bildrand)
CLIPYB	kleinste Y-Koordinate (unterer Bildrand)
CLIPYT	größte Y-Koordinate (oberer Bildrand)

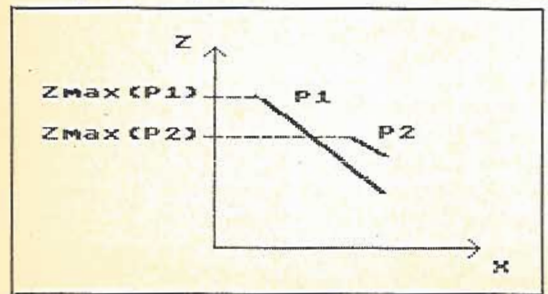


Bild 2. Fläche P1 verdeckt P2

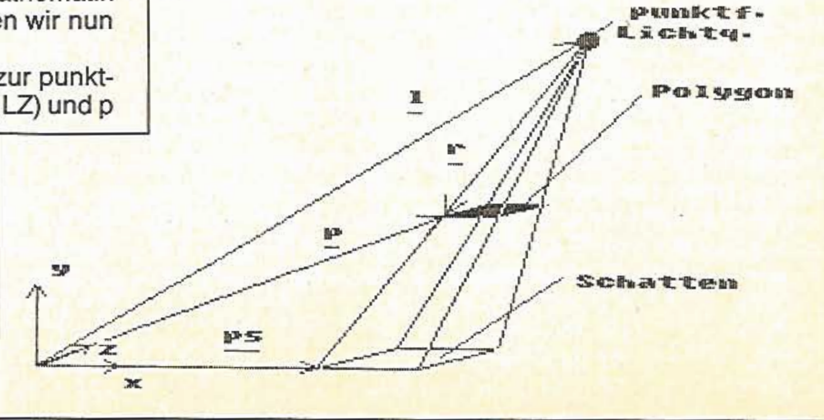


Bild 3. Berechnung und Darstellung von Schlagschatten

Im Listing ist die Einstellung eines Atari Grafikstufe 8 Screen mit 320x192 Punkten vorgegeben.

In Zeile 120 befinden sich die Koordinaten der Lichtquelle (LX,LY,LZ) und die Y-Koordinate der Schattenebene (SP).

Zur Anpassung an andere Computer ist folgendes zu beachten:

- Die Anpassung an die verschiedenen Bildschirmgrößen erfolgt mit den Variablen CLIPXL, CLIPXR, CLIPYB, CLIPYT. Diese sollten in Zeile 100 und 110 entsprechend verändert werden (zum Beispiel für Schneider oder Atari ST mit 640*400 Punkten = > CLIPXR=639, CLIPYT=399).
- Sollen mehr als 100 Polygone verwendet werden, müssen die Feldvariablen POLY_Z und PRIO entsprechend dimensioniert werden.
- Sollen mehr als 10 Eckpunkte pro Polygon verwendet werden, müssen die Feldvariablen SX, SY, SZ, X, Y entsprechend dimensioniert werden.

In letzteren beiden Fällen muß man den vorhandenen Speicherplatz berücksichtigen.

Die Transformation von Objekt- in Bildschirmkoordinaten erfolgt in der Prozedur »CALCSCREEN« in den Zeilen 800 bis 840. Die Formeln hierzu lauten:

- Zeile 810 : xc=(xv/2)*(T(0)/T(2))+(xh/2)
- Zeile 820: yc=(xv/2)*(T(1)/T(2))+(xv/2), wobei xv=maximale Y-Koordinate und xh=maximale X-Koordinate ist.

Die Transformation ist so ausgelegt, daß der Punkt 0,0 in der linken unteren Ecke liegt. Bei Computern mit dem Ursprung links oben muß der Wert yc noch einmal vom maximalen Y-Wert subtrahiert werden, um die korrekte Ansicht zu erhalten (siehe Zeile 830). Die transformierten Koordinaten befinden sich in den Variablen XC und YC.

Die diversen GRAPHICS-Befehle im Listing dienen zur Umschaltung zwischen dem Grafikmodus mit und ohne Textfenster. Auch diese Anpassung sollte keine Schwierigkeiten bereiten. Die Struktur der Prozeduren läßt sich einfach durch GOSUB und RETURN ersetzen. Gleiches gilt für die IF-, ELSE- und ENDIF-Blöcke. Probleme kann es allerdings beim UND-Operator in Zeile 3200 geben (..CA&CB...). Denn nicht jeder Computer verfügt über diese Funktion.

Zum besseren Verständnis und damit eine eventuelle Umsetzung auch gelingt, hier eine Kurzbeschreibung der einzelnen Prozeduren.

- PROC CLRMAT (480 bis 560) Löschen der Matrix MA,MB oder MC.
- PROC INITMAT (580 bis 620) Initialisieren der Matrix MA. Die Elemente der Hauptdiagonalen werden mit 1 beschrieben.
- PROC MULTMAT (640 bis 780) Multiplizieren der Matrizen MA und MB zu MC.
- PROC CALCSCREEN (800 bis 840) Transformation der dreidimensionalen in zweidimensionale Koordinaten.
- PROC CALCPOINT (860 bis 930) Multiplizieren einer Koordinate T mit der Transformationsmatrix MC.
- PROC MOVE (950 bis 1030)

Berechnen der Verschiebematrix MB zur Bewegung des Objektes. Übergabeparameter sind TX,TY und TZ.

- PROC ROTATE (1050 bis 1300)

Die Rotationsmatrix. Übergabeparameter sind ACHSE$ für die Achse und G in Gradmaß für den Winkel.

- PROC SCALE (1320 bis 1390) Skalierungsmatrix mit den Parametern SX, SY und SZ. Durch Verändern kann das Objekt in allen drei Dimensionen gestaucht oder gestreckt werden.
- PROC PERSPECTIVE (1410 bis 1460) Hier erfolgt eine perspektivische Anpassung an den Bildschirm. Durch

Verändern der Variablen wird der Fluchtpunkt verschoben. Am besten mal ausprobieren.

- PROC CLRT (1480 bis 1520)

Löschen des T-Feldes.

- PROC VIEW (1540 bis 1680)

Erstellung einer Ansicht ohne verdeckte Kanten.

Wie schon erwähnt, soll das ganze als Experimentierbasis dienen. Außerdem findet sich noch die gesondert abgedruckte Prozedur »ANIMATE« (Listing 2). Diese läßt sich mit »ENTER "D1:ANIMATE.LST"« in das Programm einbinden (Achtung: Dabei die Zeilennummern nicht verändern!). Mit Hilfe dieser Prozedur kann man Animationssequenzen zusammenstellen (Einzelbildaufnahmen).

Die Ablage der Koordinaten und Polygon-Informationen erfolgt in DATA-Zeilen. Diese werden unabhängig vom eigentlichen Programm geschrieben und dann als gesondertes File mit »LIST "D1:(NAME).IMG"« auf Diskette gespeichert. Später, nachdem das Hauptprogramm mit LOAD geladen wurde, kann das gewünschte Objekt-File mit Hilfe des ENTER-Befehls nachgeladen werden. Folgendes ist zu beachten:

- Die erste Zeilennummer ist immer 10000, und in dieser Zeile befindet sich immer die Anzahl aller Polygone des Objektes.
- Für jedes Einzelpolygon ist ein »Zehnerblock« vorgesehen, der immer bei ganzen Vielfachen von 10 beginnt (Also 10010, 10020, 10030...).
- In diesen Zeilen befindet sich dann die Anzahl der Eckpunkte des Einzelpolygons.
- Alle weiteren Zeilen enthalten die Koordinaten des Polygons. Hierbei erfolgt die Numerierung in Einserschritten.
- Zuletzt sollte noch eine Abschlußzeile eingefügt werden, die den Wert 0 enthält.

Ein einfaches quadratisches Polygon müßte so definiert werden:

10020 DATA 4: Ein Quadrat hat bekanntlich 4 Eckpunkte
10021 DATA X1,Y1,Z1: Koordinate des 1. Eckpunktes
10022 DATA X2,Y2,Z2: Koordinate des 2. Eckpunktes
10023 DATA X3,Y3,Z3: Koordinate des 3. Eckpunktes
10024 DATA X4,Y4,Z4: Koordinate des 4. Eckpunktes
10030 DATA.....(nächstes Polygon)

Auf diese Art und Weise ensteht eine bestimmte Ordnung in der Zeilennumerierung. So würde zum Beispiel die Zeilennummer 10232 entsprechend der letzten drei Ziffern besagen: 23. Polygon, 2. Eckpunkt. Hierzu finden sich auch drei Beispiele abgedruckt (Listing 3 bis 5).

Die Bedienung des Programms ist recht simpel. Folgende Kommandos stehen zur Verfügung:
MOVE, ROTATE, SCALE, VIEW und HIDE.

Die Parameter X, Y und Z beziehen sich bei den Befehlen MOVE und SCALE auf Koordinaten, beim ROTATE-Kommando sind die Winkel für jede Achse einzugeben.

Eines ist unbedingt zu beachten: Nachdem alle Bewegungs- und Rotationskommandos ausgeführt sind, muß für die korrekte Darstellung immer die Anpassungsprozedur »PERSPECTIVE« aufgerufen werden. Diese gewährleistet die korrekte Positionierung des Fluchtpunktes. Eine Kommandoeingabe kann so aussehen:

command: ROTATE (RETURN) :Der Rotationsbefehl
X,Y,Y: 10,20,0 (RETURN) :Rotiere 10 Grad über X und 20 Grad über Y
command: MOVE (RETURN) :Der Move Befehl
X,Y,Z: 0,0,10 (RETURN) :Bewegung um 10 Einheiten in Z-Richtung
command: PERSPECTIVE (RETURN) :Perspektivische Anpassung
command: VIEW :Darstellung als Drahtmodell
command: HIDE :Darstellung mit Schlagschatten und verdeckten Kanten.

(Theo Janssen/rz/hf)

```
10 REM {CTL Q}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL E}                          <LD>
20 REM {SHIFT =}        3D GRAFIK PAKET
   {SHIFT =}                               <EG>
30 REM {SHIFT =}  BY THEO JANSSEN 1986/8
7   {SHIFT =}                              <FR>
40 REM {CTL Z}{CTL R}{CTL R}{CTL R}{CTL
R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CT
L R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{
CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R
}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL
 R}{CTL R}{CTL C}                          <LE>
50 REM                                     <JS>
60 REM                                     <JT>
70 DIM MA(3,3),MB(3,3),MC(3,3)             <LA>
80 DIM P(3),T(3),POLY_Z(100),PRIO(100),S
X(10),SY(10),SZ(10)                        <PH>
90 DIM ACHSE$(1),KOM$(20),X(10),Y(10)      <XM>
100 CLIPXL=0:CLIPXR=319                    <DK>
110 CLIPYB=0:CLIPYT=191                    <AF>
120 LX=0:LY=70:LZ=0:SP=-10                 <FD>
130 GRAPHICS 8:EXEC SET_COL                <HJ>
140 COLOR 1                                <XR>
150 EXEC INITMAT                           <DS>
160 INPUT "command>";KOM$                  <KG>
170 IF KOM$(1,3)="VIE" THEN CLS #6:EXEC
VIEW                                       <MO>
180 IF KOM$(1,3)="INI" THEN EXEC INITMAT   <MH>
190 IF KOM$(1,3)="PER" THEN EXEC PERSPEC
TIVE                                       <ZH>
200 IF KOM$(1,3)="MOV"                     <EE>
210 INPUT "X,Y,Z:>";AA,AB,AC               <IN>
220 TX=AA:TY=AB:TZ=AC                      <RT>
230 EXEC MOVE                              <DK>
240 ENDIF                                  <UZ>
250 IF KOM$(1,3)="ROT"                     <GP>
260 INPUT "X,Y,Z:>";AA,AB,AC               <IX>
270 ACHSE$="X":G=AA:EXEC ROTATE            <BG>
280 ACHSE$="Y":G=AB:EXEC ROTATE            <CO>
290 ACHSE$="Z":G=AC:EXEC ROTATE            <DW>
300 ENDIF                                  <US>
310 IF KOM$(1,3)="SCA"                     <IM>
320 INPUT "X,Y,Z:>";AA,AB,AC               <IQ>
330 SX=AA:SY=AB:SZ=AC                      <QP>
340 EXEC SCALE                             <FS>
350 ENDIF                                  <VC>
360 IF KOM$(1,3)="HID"                     <HW>
370 GRAPHICS 24                            <XY>
380 EXEC SET_COL                           <MK>
390 EXEC HIDDENVIEW                        <YU>
400 GET KEY                                <MJ>
410 GRAPHICS 40                            <WH>
420 EXEC SET_COL                           <LZ>
430 ENDIF                                  <UZ>
440 IF KOM$(1,3)="LIG" THEN INPUT "NEW L
IGHT POSITION (X,Y,Z):";LX,LY,LZ           <MC>
450 IF KOM$(1,3)="SHA" THEN INPUT "SHADO
W(S) ON Y-HEIGHT:";SP                      <WP>
460 GOTO 160                               <OJ>
470 --                                     <JT>
480 PROC CLRMAT                            <JU>
490 FOR I=0 TO 3                           <MR>
500 FOR J=0 TO 3                           <MJ>
510 IF MAT=1 THEN MA(J,I)=0                <AE>
520 IF MAT=2 THEN MB(J,I)=0                <BM>
530 IF MAT=3 THEN MC(J,I)=0                <CU>
540 NEXT J                                 <GN>
550 NEXT I                                 <GF>
560 ENDPROC                                <KT>
570 --                                     <JU>
580 PROC INITMAT                           <HN>
590 MAT=1:EXEC CLRMAT                      <KA>
```

```
600 MA(0,0)=1:MA(1,1)=1                       <CY>
610 MA(2,2)=1:MA(3,3)=1                       <HI>
620 ENDPROC                                   <KM>
630 --                                        <JN>
640 PROC MULTMAT                              <ND>
650 MAT=3:EXEC CLRMAT                         <KL>
660 FOR K=0 TO 3                              <NF>
670 FOR J=0 TO 3                              <MY>
680 FOR I=0 TO 3                              <MR>
690 MC(J,K)=MC(J,K)+MA(I,K)*MB(J,I)           <LQ>
700 NEXT I                                    <FX>
710 NEXT J                                    <GJ>
720 NEXT K                                    <GV>
730 FOR I=0 TO 3                              <MI>
740 FOR J=0 TO 3                              <MT>
750 MA(I,J)=MC(I,J)                           <FL>
760 NEXT J                                    <GT>
770 NEXT I                                    <GL>
780 ENDPROC                                   <KZ>
790 --                                        <KA>
800 PROC CALCSCREEN                           <PD>
810 XC=95.5*(T(0)/T(2))+159.5                 <QY>
820 YC=95.5*(T(1)/T(2))+95.5                  <ON>
830 YC=191-YC                                 <EV>
840 ENDPROC                                   <KS>
850 --                                        <JT>
860 PROC CALCPOINT                            <AQ>
870 FOR J=0 TO 3                              <NA>
880 FOR I=0 TO 3                              <MT>
890 T(J)=T(J)+P(I)*MC(J,I)                    <LD>
900 NEXT I                                    <FZ>
910 NEXT J                                    <GL>
920 IF T(2)<1 THEN T(2)=1                     <TF>
930 ENDPROC                                   <KR>
940 --                                        <JS>
950 PROC MOVE                                 <HG>
960 MAT=2:EXEC CLRMAT                         <KH>
970 MB(0,0)=1:MB(1,1)=1                       <EL>
980 MB(2,2)=1:MB(3,3)=1                       <IV>
990 MB(0,3)=TX                                <IJ>
1000 MB(1,3)=TY                               <NZ>
1010 MB(2,3)=TZ                               <PA>
1020 EXEC MULTMAT                             <ZQ>
1030 ENDPROC                                  <OL>
1040 --                                       <VQ>
1050 PROC ROTATE                              <AR>
1060 DEG                                      <OO>
1070 MAT=2:EXEC CLRMAT                        <LZ>
1080 IF ACHSE$="X"                            <PG>
1090 MB(0,0)=1:MB(3,3)=1                      <DC>
1100 MB(1,1)=COS(G)                           <YQ>
1110 MB(2,1)=-SIN(G)                          <OG>
1120 MB(1,2)=SIN(G)                           <BJ>
1130 MB(2,2)=COS(G)                           <ZT>
1140 ENDIF                                    <SU>
1150 IF ACHSE$="Y"                            <PQ>
1160 MB(1,1)=1:MB(3,3)=1                      <DP>
1170 MB(0,0)=COS(G)                           <YR>
1180 MB(2,0)=SIN(G)                           <BO>
1190 MB(0,2)=-SIN(G)                          <OX>
1200 MB(2,2)=COS(G)                           <ZM>
1210 ENDIF                                    <SN>
1220 IF ACHSE$="Z"                            <QA>
1230 MB(2,2)=1:MB(3,3)=1                      <EC>
1240 MB(0,0)=COS(G)                           <YK>
1250 MB(1,0)=-SIN(G)                          <OA>
1260 MB(0,1)=SIN(G)                           <BD>
1270 MB(1,1)=COS(G)                           <ZN>
1280 ENDIF                                    <TI>
1290 EXEC MULTMAT                             <AP>
1300 ENDPROC                                  <OI>
1310 --                                       <VN>
1320 PROC SCALE                               <ST>
1330 MAT=2:EXEC CLRMAT                        <LT>
1340 MB(3,3)=1                                <XE>
1350 MB(0,0)=SX                               <MB>
1360 MB(1,1)=SY                               <NN>
1370 MB(2,2)=SZ                               <OZ>
1380 EXEC MULTMAT                             <AO>
1390 ENDPROC                                  <PJ>
1400 --                                       <VM>
1410 PROC PERSPECTIVE                         <SQ>
1420 MAT=2:EXEC CLRMAT                        <LS>
1430 MB(0,0)=4:MB(1,1)=4                      <EA>
1440 MB(2,2)=1:MB(3,3)=1                      <EJ>
1450 EXEC MULTMAT                             <AH>
1460 ENDPROC                                  <PC>
1470 --                                       <WH>
1480 PROC CLRT                                <PX>
1490 FOR I=0 TO 3                             <XQ>
1500 T(I)=0                                   <FQ>
1510 NEXT I                                   <FC>
1520 ENDPROC                                  <OS>
1530 --                                       <VX>
1540 PROC VIEW                                <RH>
1550 RESTORE (10000)                          <ZG>
1560 READ P_NUMBER                            <AM>
1570 FOR K=1 TO P_NUMBER                      <SP>
1580 READ E_NUMBER,XP,YP,ZP                   <TS>
1590 XA=XP:YA=YP:ZA=ZP                        <IM>
1600 FOR L=2 TO E_NUMBER                      <LN>
1610 READ XL,YL,ZL                            <BF>
1620 EXEC SHOWLINE                            <VY>
1630 XP=XL:YP=YL:ZP=ZL                        <WZ>
1640 NEXT L                                   <GU>
1650 XL=XA:YL=YA:ZL=ZA                        <WX>
1660 EXEC SHOWLINE                            <WK>
1670 NEXT K                                   <GS>
1680 ENDPROC                                  <PM>
1690 --                                       <WR>
1700 PROC SORTPOLY                            <VL>
1710 RESTORE (10000)                          <YY>
1720 READ P_NUMBER                            <AE>
1730 FOR R=1 TO P_NUMBER                      <UZ>
1740 POLY_Z(R)=0                              <CM>
1750 NEXT R                                   <JN>
1760 TEXT 0,180,"sorting▲Polygons▲:"          <YU>
1770 FOR R=1 TO P_NUMBER                      <VL>
1780 TEXT 152,180,R                           <UY>
1790 ZGES=0                                   <DW>
1800 READ E_NUMBER                            <VJ>
1810 FOR S=1 TO E_NUMBER                      <OA>
1820 READ X,Y,Z                               <KN>
1830 P(0)=X:P(1)=Y:P(2)=Z:P(3)=1              <WW>
1840 EXEC CLRT                                <LT>
1850 EXEC CALCPOINT                           <SN>
1860 ZGES=ZGES+T(2)                           <OO>
1870 IF T(2)>POLY_Z(R)                        <UA>
1880 POLY_Z(R)=T(2)                           <LX>
1890 ENDIF                                    <TX>
1900 NEXT S                                   <JN>
1910 ZMIT=ZGES/E_NUMBER                       <DH>
1920 POLY_Z(R)=POLY_Z(R)+ZMIT                 <HA>
1930 NEXT R                                   <JL>
1940 FOR R=1 TO P_NUMBER                      <VG>
1950 PRIO(R)=1                                <IE>
1960 NEXT R                                   <JU>
1970 FOR S=1 TO P_NUMBER                      <VZ>
1980 W=1                                      <ZW>
1990 FOR R=1 TO P_NUMBER                      <VV>
2000 IF (POLY_Z(R)>=POLY_Z(W)) THEN W=R       <AR>
2010 NEXT R                                   <IO>
2020 PRIO(S)=W                                <CB>
2030 POLY_Z(W)=-9E-20                         <PR>
2040 NEXT S                                   <JI>
2050 ENDPROC                                  <OS>
2060 --                                       <VX>
2070 PROC HIDDENVIEW                          <AO>
2080 EXEC SORTPOLY                            <RW>
2090 CLS #6                                   <EP>
2100 RESTORE (10000)                          <YK>
2110 READ P_NUMBER                            <ZQ>
2120 FOR Q=1 TO P_NUMBER                      <UB>
2130 RESTORE (10000+(PRIO(Q)*10))             <TO>
2140 READ E_NUMBER                            <VI>
2150 FOR S=1 TO E_NUMBER                      <NZ>
2160 READ X,Y,Z                               <KM>
2170 P(0)=X:P(1)=Y:P(2)=Z:P(3)=1              <WV>
2180 EXEC CLRT                                <LS>
2190 EXEC CALCPOINT                           <SM>
2200 EXEC CALCSHADOW                          <OB>
```

Listing 1. »3D-Grafik«

```
2210 EXEC CALCSCREEN                  <IR>
2220 X(S)=XC:Y(S)=YC                  <EF>
2230 NEXT S                           <JJ>
2240 FOR S=1 TO E_NUMBER              <NY>
2250 T=S+1                            <QN>
2260 IF T>E_NUMBER THEN T=1           <CM>
2270 XH=X(S):YH=Y(S)                  <YC>
2280 XJ=X(T):YJ=Y(T)                  <BB>
2290 CL=0:EXEC DDA                    <TR>
2300 NEXT S                           <JC>
2310 FOR S=1 TO E_NUMBER              <NR>
2320 T=S+1                            <QG>
2330 IF T>E_NUMBER THEN T=1           <CF>
2340 XF=X(S):YF=Y(S)                  <WD>
2350 XG=X(T):YG=Y(T)                  <YG>
2360 COLOR 1                          <YU>
2370 EXEC CLIP                        <FC>
2380 IF VIS                           <IJ>
2390 PLOT XF,YF:DRAWTO XG,YG          <VE>
2400 ENDIF                            <SP>
2410 NEXT S                           <JH>
2420 EXEC PLOTSHADOW                  <JR>
2430 NEXT Q                           <IR>
2440 ENDPROC                          <OX>
2450 --                               <WC>
2460 PROC DDA                         <KS>
2470 LENGTH=ABS(XJ-XH)                <UE>
2480 IF ABS(YJ-YH)>LENGTH THEN LENGTH=AB
S(YJ-YH)                              <DF>
2490 XINC=(XJ-XH)/LENGTH              <CI>
2500 YINC=(YJ-YH)/LENGTH              <CQ>
2510 XD=XH+0.5:YD=YH+0.5              <QY>
2520 FOR POINT=1 TO LENGTH            <FA>
2530 FOR N=1 TO E_NUM                 <BO>
2540 COLOR CL                         <TA>
2550 XF=XD:YF=YD:XG=X(N):YG=Y(N)      <XJ>
2560 EXEC CLIP                        <FD>
2570 IF VIS                           <IK>
2580 PLOT XF,YF                       <YM>
2590 DRAWTO XG,YG                     <NI>
2600 PLOT INT(XF),INT(YF)             <DD>
2610 DRAWTO XG,YG                     <MM>
2620 ENDIF                            <SZ>
2630 NEXT N                           <HO>
2640 XD=XD+XINC                       <DP>
2650 YD=YD+YINC                       <ET>
2660 NEXT POINT                       <QE>
2670 ENDPROC                          <PK>
2680 --                               <WP>
2690 PROC SHOWLINE                    <AU>
2700 EXEC CLRT                        <LG>
2710 P(0)=XP:P(1)=YP:P(2)=ZP:P(3)=1   <NL>
2720 EXEC CALCPOINT                   <SD>
2730 EXEC CALCSCREEN                  <JH>
2740 XF=XC:YF=YC                      <BX>
2750 EXEC CLRT                        <LV>
2760 P(0)=XL:P(1)=YL:P(2)=ZL:P(3)=1   <EU>
2770 EXEC CALCPOINT                   <SS>
2780 EXEC CALCSCREEN                  <JW>
2790 XG=XC:YG=YC                      <DG>
2800 EXEC CLIP                        <ER>
2810 IF VIS                           <HY>
2820 PLOT XF,YF                       <YA>
2830 DRAWTO XG,YG                     <MW>
2840 ENDIF                            <TJ>
2850 ENDPROC                          <PI>
2860 --                               <WN>
2870 PROC SET_COL                     <HO>
2880 SETCOLOR 2,0,12                  <UH>
2890 SETCOLOR 4,0,12                  <VO>
2900 SETCOLOR 1,0,0                   <FR>
2910 ENDPROC                          <OY>
2920 --                               <WD>
2930 PROC CODE                        <CB>
2940 C=0                              <UN>
2950 IF X<CLIPXL                      <BP>
2960 C=8                              <XF>
2970 ELSE                             <UI>
2980 IF X>CLIPXR                      <GK>
2990 C=4                              <WI>
3000 ENDIF                            <SI>
3010 ENDIF                            <SL>
3020 IF Y<CLIPYB                      <VJ>
3030 C=C+2                            <HS>
3040 ELSE                             <TI>
3050 IF Y>CLIPYT                      <HO>
3060 C=C+1                            <HR>
3070 ENDIF                            <TD>
3080 ENDIF                            <TG>
3090 ENDPROC                          <PF>
3100 --                               <VI>
3110 PROC CLIP                        <IH>
3120 VIS=1                            <AB>
3130 X=XF:Y=YF                        <XN>
3140 EXEC CODE                        <XP>
3150 CA=C                             <WI>
3160 X=XG:Y=YG                        <YT>
3170 EXEC CODE                        <XY>
3180 CB=C                             <WY>
3190 WHILE (CA<>0) OR (CB<>0)         <XQ>
3200 IF (CA&CB)<>0 THEN LET VIS=0:ENDPRO
C                                     <WT>
3210 C=CA:CZ=CA                       <KH>
3220 IF C=0 THEN LET C=CB:CZ=CB       <XX>
3230 IF (C/8)>=1                      <PJ>
3240 Y=YF+(YG-YF)*(CLIPXL-XF)/(XG-XF) <BY>
3250 X=CLIPXL                         <PR>
3260 C=C-8                            <LF>
3270 ENDIF                            <TH>
3280 IF (C/8)>=0.5                    <EC>
3290 Y=YF+(YG-YF)*(CLIPXR-XF)/(XG-XF) <IH>
3300 X=CLIPXR                         <SE>
3310 C=C-4                            <JE>
3320 ENDIF                            <SU>
3330 IF (C/8)>=0.25                   <OG>
3340 X=XF+(XG-XF)*(CLIPYB-YF)/(YG-YF) <VJ>
3350 Y=CLIPYB                         <LL>
3360 C=C-2                            <IZ>
3370 ENDIF                            <TJ>
3380 IF (C/8)>=0.125                  <AS>
3390 X=XF+(XG-XF)*(CLIPYT-YF)/(YG-YF) <NG>
3400 Y=CLIPYT                         <TY>
3410 C=C-1                            <IC>
3420 ENDIF                            <SW>
3430 IF CZ=CA                         <JZ>
3440 XF=X:YF=Y:EXEC CODE              <EF>
3450 CA=C                             <WO>
3460 ELSE                             <TW>
3470 XG=X:YG=Y:EXEC CODE              <FH>
3480 CB=C                             <XE>
3490 ENDIF                            <TR>
3500 WEND                             <TQ>
3510 ENDPROC                          <OR>
3520 --                               <VW>
3530 PROC CALCSHADOW                  <SR>
3540 SX(S)=LX-T(0)                    <XF>
3550 SY(S)=LY-T(1)                    <YT>
3560 SZ(S)=LZ-T(2)                    <AH>
3570 SKALAR=-((LY+(-SP))/SY(S))       <IL>
3580 SX(S)=LX+SX(S)*SKALAR            <UI>
3590 SY(S)=SP                         <WF>
3600 SZ(S)=LZ+SZ(S)*SKALAR            <WG>
3610 ENDPROC                          <OT>
3620 --                               <VY>
3630 PROC PLOTSHADOW                  <NZ>
3640 FOR S=1 TO E_NUMBER              <OH>
3650 T(0)=SX(S):T(1)=SY(S)            <IB>
3660 T(2)=SZ(S):T(3)=1                <FY>
3670 EXEC CALCSCREEN                  <JS>
3680 X(S)=XC:Y(S)=YC                  <FG>
3690 NEXT S                           <KK>
3700 FOR S=1 TO E_NUMBER              <NX>
3710 T=S+1                            <QM>
3720 IF T>E_NUMBER THEN T=1           <CL>
3730 XH=X(S):YH=Y(S)                  <YB>
3740 XJ=X(T):YJ=Y(T)                  <BA>
3750 CL=1:EXEC DDA                    <TZ>
3760 NEXT S                           <KD>
3770 ENDPROC                          <PN>
3780 --                               <WS>
```

Listing 1. »3D-Grafik« (Schluß)



```
455 IF KOM$(1,3)="ANI" THEN EXEC ANIMATE <AJ>
5000 PROC ANIMATE                        <KY>
5010 GRAPHICS 0:EXEC SET_COL             <VP>
5020 PRINT "STARTPOSITION"               <RS>
5030 INPUT "Winkel▲(x,y,z)";X_WS,Y_WS,Z_
WS                                       <GY>
5040 INPUT "Position▲(x,y,z)";X_PS,Y_PS,
Z_PS                                     <EF>
5050 PRINT :PRINT :PRINT "ENDPOSITION"   <OM>
5060 INPUT "Winkel▲(x,y,z)";X_WE,Y_WE,Z_
WE                                       <ZN>
5070 INPUT "Position▲(x,y,z)";X_PE,Y_PE,
Z_PE                                     <TO>
5080 PRINT :PRINT :PRINT                 <QY>
5090 INPUT "in▲welchem▲Zeitraum▲(sec.)";
TI                                       <CR>
5100 GRAPHICS 24:EXEC SET_COL            <MS>
5110 STEP=TI*25                          <AX>
5120 WMX=(X_WE-X_WS)/STEP                <ZA>
5130 WMY=(Y_WE-Y_WS)/STEP                <AM>
5140 WMZ=(Z_WE-Z_WS)/STEP                <BY>
5150 PMX=(X_PE-X_PS)/STEP                <PK>
5160 PMY=(Y_PE-Y_PS)/STEP                <QW>
5170 PMZ=(Z_PE-Z_PS)/STEP                <SI>
5180 FOR BILD=0 TO STEP                  <WZ>
5190 EXEC INITMAT                        <UG>
5200 G=X_WS:ACHSE$="X":EXEC ROTATE       <NE>
5210 G=Y_WS:ACHSE$="Y":EXEC ROTATE       <OK>
5220 G=Z_WS:ACHSE$="Z":EXEC ROTATE       <PQ>
5230 TX=X_PS:TY=Y_PS:TZ=Z_PS             <VO>
5240 EXEC MOVE                           <KZ>
5250 EXEC PERSPECTIVE                    <PC>
5260 EXEC HIDDENVIEW                     <WS>
5270 GET KEY                             <OU>
5280 X_WS=X_WS+WMX                       <PQ>
5290 Y_WS=Y_WS+WMY                       <RC>
5300 Z_WS=Z_WS+WMZ                       <RM>
5310 X_PS=X_PS+PMX                       <EY>
5320 Y_PS=Y_PS+PMY                       <GK>
5330 Z_PS=Z_PS+PMZ                       <HW>
5340 CLS #6                              <EJ>
5350 NEXT BILD                           <GX>
5360 ENDPROC                             <PE>
5370 --                                  <WJ>
```

Listing 2. »ANIMATE.LST«

```
10000 DATA▲6                <JZ>
10010 DATA▲4                <JF>
10011 DATA▲-1,1,-1          <RB>
10012 DATA▲1,1,-1           <LS>
10013 DATA▲1,-1,-1          <RI>
10014 DATA▲-1,-1,-1         <YU>
10020 DATA▲4                <JJ>
10021 DATA▲1,1,1            <IC>
10022 DATA▲-1,1,1           <MC>
10023 DATA▲-1,-1,1          <RS>
10024 DATA▲1,-1,1           <MJ>
10030 DATA▲4                <JN>
10031 DATA▲-1,1,1           <MB>
10032 DATA▲-1,1,-1          <RO>
10033 DATA▲-1,-1,-1         <YX>
10034 DATA▲-1,-1,1          <SB>
10040 DATA▲4                <JR>
10041 DATA▲1,1,-1           <LZ>
10042 DATA▲1,1,1            <IP>
10043 DATA▲1,-1,1           <MM>
10044 DATA▲1,-1,-1          <RZ>
10050 DATA▲4                <JV>
10051 DATA▲-1,1,1           <MJ>
10052 DATA▲1,1,1            <IT>
10053 DATA▲1,1,-1           <MN>
10054 DATA▲-1,1,-1          <SG>
10060 DATA▲4                <JZ>
10061 DATA▲1,-1,1           <MK>
10062 DATA▲-1,-1,1          <SD>
10063 DATA▲-1,-1,-1         <ZJ>
10064 DATA▲1,-1,-1          <SH>
20000 DATA▲0                <HG>
```

Listing 3. »WUERFEL.IMG«

```
10000 DATA▲19               <KE>
10010 DATA▲4                <JF>
10011 DATA▲-1,1,-1          <RB>
10012 DATA▲1,1,-1           <LS>
10013 DATA▲1,-1,-1          <RI>
10014 DATA▲-1,-1,-1         <YU>
10020 DATA▲4                <JJ>
10021 DATA▲1,1,1            <IC>
10022 DATA▲-1,1,1           <MC>
10023 DATA▲-1,-1,1          <RS>
10024 DATA▲1,-1,1           <MJ>
10030 DATA▲4                <JN>
10031 DATA▲-1,1,1           <MB>
10032 DATA▲-1,1,-1          <RO>
10033 DATA▲-1,-1,-1         <YX>
10034 DATA▲-1,-1,1          <SB>
10040 DATA▲4                <JR>
10041 DATA▲1,1,-1           <LZ>
10042 DATA▲1,1,1            <IP>
10043 DATA▲1,-1,1           <MM>
10044 DATA▲1,-1,-1          <RZ>
10050 DATA▲4                <JV>
10051 DATA▲-1,1,1           <MJ>
10052 DATA▲1,1,1            <IT>
10053 DATA▲1,1,-1           <MN>
10054 DATA▲-1,1,-1          <SG>
10060 DATA▲4                <JZ>
10061 DATA▲1,-1,1           <MK>
10062 DATA▲-1,-1,1          <SD>
10063 DATA▲-1,-1,-1         <ZJ>
10064 DATA▲1,-1,-1          <SH>
10070 DATA▲4                <KD>
10071 DATA▲2,0,2            <JK>
10072 DATA▲4,0,2            <KN>
10073 DATA▲4,0,4            <LY>
10074 DATA▲2,0,4            <LF>
10080 DATA▲3                <JV>
10081 DATA▲2,0,2            <JO>
10082 DATA▲3,2,3            <LX>
10083 DATA▲4,0,2            <KW>
10090 DATA▲3                <JZ>
10091 DATA▲2,0,4            <KY>
10092 DATA▲3,2,3            <MB>
10093 DATA▲4,0,4            <MG>
10100 DATA▲3                <IS>
10101 DATA▲2,0,2            <IL>
10102 DATA▲3,2,3            <KU>
10103 DATA▲2,0,4            <KB>
10110 DATA▲3                <IW>
10111 DATA▲4,0,2            <JN>
10112 DATA▲3,2,3            <KY>
10113 DATA▲4,0,4            <LD>
10120 DATA▲3                <JA>
10121 DATA▲-2,0,-2          <RY>
10122 DATA▲-3,2,-3          <UM>
10123 DATA▲-4,0,-2          <TI>
10130 DATA▲3                <JE>
10131 DATA▲-2,0,-4          <TM>
10132 DATA▲-3,2,-3          <UQ>
10133 DATA▲-4,0,-4          <UW>
10140 DATA▲3                <JI>
10141 DATA▲-2,0,-2          <SG>
10142 DATA▲-3,2,-3          <UU>
10143 DATA▲-2,0,-4          <UA>
10150 DATA▲3                <JM>
10151 DATA▲-4,0,-2          <TK>
```

Listing 4. »KOERPER.IMG«

```
10152 DATA -3,2,-3          <UY>
10153 DATA -4,0,-4          <VE>
10160 DATA 3                <JQ>
10161 DATA -2,0,-2          <SO>
10162 DATA -3,-2,-3         <CJ>
10163 DATA -4,0,-2          <TY>
10170 DATA 3                <JU>
10171 DATA -2,0,-4          <UC>
10172 DATA -3,-2,-3         <CN>
10173 DATA -4,0,-4          <VM>
10180 DATA 3                <JY>
10181 DATA -2,0,-2          <SW>
10182 DATA -3,-2,-3         <CR>
10183 DATA -2,0,-4          <UQ>
10190 DATA 3                <KC>
10191 DATA -4,0,-2          <UA>
10192 DATA -3,-2,-3         <CV>
10193 DATA -4,0,-4          <VU>
20000 DATA 0                <HG>
```

Listing 4. »KOERPER.IMG« (Schluß)

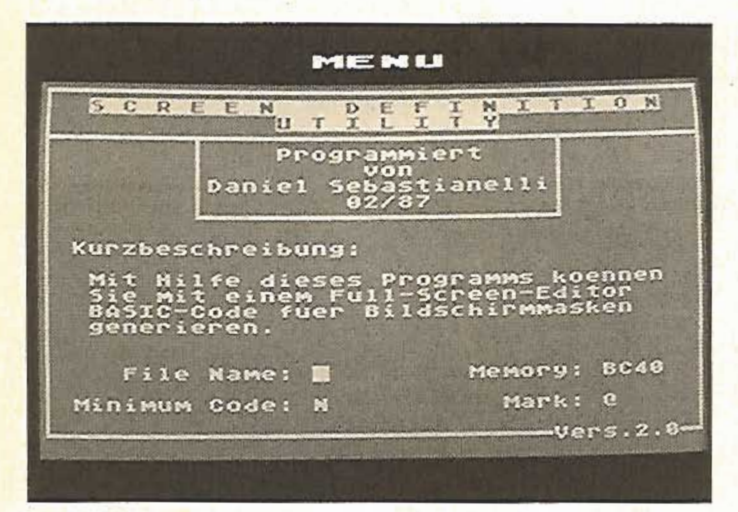
```
10000 DATA 26               <JD>
10010 DATA 4                <JF>
10011 DATA -.2,-1.4,-.2     <AK>
10012 DATA -.2,1.4,-.2      <MD>
10013 DATA .2,1.4,-.2       <ZK>
10014 DATA .2,-1.4,-.2      <MI>
10020 DATA 4                <JJ>
10021 DATA -.2,-1.4,.2      <MN>
10022 DATA -.2,1.4,.2       <ZZ>
10023 DATA .2,1.4,.2        <OZ>
10024 DATA .2,-1.4,.2       <AE>
10030 DATA 4                <JN>
10031 DATA -.2,-1.4,-.2     <AS>
10032 DATA .2,-1.4,-.2      <MG>
10033 DATA .2,-1.4,.2       <AD>
10034 DATA -.2,-1.4,.2      <NG>
10040 DATA 4                <JR>
10041 DATA -.2,1.4,-.2      <MK>
10042 DATA .2,1.4,-.2       <ZR>
10043 DATA .2,1.4,.2        <PH>
10044 DATA -.2,1.4,.2       <AR>
10050 DATA 4                <JV>
10051 DATA -.2,-1.4,.2      <MZ>
10052 DATA -.2,1.4,.2       <AL>
10053 DATA -.2,1.4,-.2      <MY>
10054 DATA -.2,-1.4,-.2     <BP>
10060 DATA 4                <JZ>
10061 DATA .2,-1.4,.2       <AF>
10062 DATA .2,1.4,.2        <PK>
10063 DATA .2,1.4,-.2       <AE>
10064 DATA .2,-1.4,-.2      <NC>
10070 DATA 4                <KD>
10071 DATA -.8,1.4,-.2      <QC>
10072 DATA -.4,1.4,-.2      <OD>
10073 DATA -.4,-.2,-.2      <KM>
10074 DATA -.8,-.2,-.2      <MV>
10080 DATA 4                <KH>
10081 DATA -.8,1.4,.2       <DY>
10082 DATA -.4,1.4,.2       <BZ>
10083 DATA -.4,-.2,.2       <YI>
10084 DATA -.8,-.2,.2       <AR>
10090 DATA 4                <KL>
10091 DATA -.8,1.4,-.2      <QK>
10092 DATA -.8,1.4,.2       <EH>
10093 DATA -.4,1.4,.2       <CI>
10094 DATA -.4,1.4,-.2      <OV>
10100 DATA 4                <JE>
10101 DATA -.4,1.4,-.2      <MZ>
10102 DATA -.4,1.4,.2       <AW>
10103 DATA -.4,-.2,.2       <XF>
10104 DATA -.4,-.2,-.2      <JS>
10110 DATA 4                <JI>
10111 DATA -.8,1.4,-.2      <PH>
10112 DATA -.8,1.4,.2       <DE>
10113 DATA -.8,-.2,.2       <ZW>
10114 DATA -.8,-.2,-.2      <MA>
10120 DATA 4                <JM>
10121 DATA .8,1.4,-.2       <CH>
10122 DATA .4,1.4,-.2       <AM>
10123 DATA .4,-.2,-.2       <XB>
10124 DATA .8,-.2,-.2       <ZG>
10130 DATA 4                <JQ>
10131 DATA .8,1.4,.2        <RW>
10132 DATA .4,1.4,.2        <QB>
10133 DATA .4,-.2,.2        <MQ>
10134 DATA .8,-.2,.2        <OV>
10140 DATA 4                <JU>
10141 DATA .8,1.4,-.2       <CP>
10142 DATA .8,1.4,.2        <SF>
10143 DATA .4,1.4,.2        <QK>
10144 DATA .4,1.4,-.2       <BE>
10150 DATA 4                <JY>
10151 DATA .4,1.4,-.2       <AT>
10152 DATA .4,1.4,.2        <QJ>
10153 DATA .4,-.2,.2        <MY>
10154 DATA .4,-.2,-.2       <XS>
10160 DATA 4                <KC>
10161 DATA .8,1.4,-.2       <CX>
10162 DATA .8,1.4,.2        <SN>
10163 DATA .8,-.2,.2        <PC>
10164 DATA .8,-.2,-.2       <ZW>
10170 DATA 4                <KG>
10171 DATA -.8,-.2,-.2      <MJ>
10172 DATA -.4,-.2,-.2      <KK>
10173 DATA -1.2,-1.4,-.2    <UF>
10174 DATA -1.8,-1.4,-.2    <XW>
10180 DATA 4                <KK>
10181 DATA -.8,-.2,.2       <AF>
10182 DATA -.4,-.2,.2       <YG>
10183 DATA -1.2,-1.4,.2     <EP>
10184 DATA -1.8,-1.4,.2     <IG>
10190 DATA 4                <KO>
10191 DATA -.8,-.2,.2       <AJ>
10192 DATA -1.8,-1.4,.2     <IA>
10193 DATA -1.8,-1.4,-.2    <XZ>
10194 DATA -.8,-.2,-.2      <NG>
10200 DATA 4                <JH>
10201 DATA -.4,-.2,.2       <WY>
10202 DATA -1.2,-1.4,.2     <DH>
10203 DATA -1.2,-1.4,-.2    <TG>
10204 DATA -.4,-.2,-.2      <JV>
10210 DATA 4                <JL>
10211 DATA -1.8,-1.4,-.2    <WM>
10212 DATA -1.2,-1.4,-.2    <TF>
10213 DATA -1.2,-1.4,.2     <DQ>
10214 DATA -1.8,-1.4,.2     <HH>
10220 DATA 4                <JP>
10221 DATA .8,-.2,-.2       <YU>
10222 DATA .4,-.2,-.2       <WZ>
10223 DATA 1.2,-1.4,-.2     <DA>
10224 DATA 1.8,-1.4,-.2     <GL>
10230 DATA 4                <JT>
10231 DATA .8,-.2,.2        <OJ>
10232 DATA .4,-.2,.2        <MO>
10233 DATA 1.2,-1.4,.2      <PD>
10234 DATA 1.8,-1.4,.2      <SO>
10240 DATA 4                <JX>
10241 DATA .8,-.2,.2        <ON>
10242 DATA 1.8,-1.4,.2      <SI>
10243 DATA 1.8,-1.4,-.2     <GO>
10244 DATA .8,-.2,-.2       <ZR>
10250 DATA 4                <KB>
10251 DATA .4,-.2,.2        <MR>
10252 DATA 1.2,-1.4,.2      <PG>
10253 DATA 1.2,-1.4,-.2     <DM>
10254 DATA .4,-.2,-.2       <XV>
10260 DATA 4                <KF>
10261 DATA 1.8,-1.4,-.2     <GM>
10262 DATA 1.2,-1.4,-.2     <DL>
10263 DATA 1.2,-1.4,.2      <PP>
10264 DATA 1.8,-1.4,.2      <TA>
20000 DATA 0                <HG>
```

Listing 5. »LOGO.IMG«

Benutzerfreundliche Programme

Um ein Programm benutzerfreundlich zu gestalten, bedarf es viel Zeit. Zeit, die eigentlich besser genutzt werden könnte. Mit unserem Programm sparen Sie diese Zeit, denn Menüsteuerungen sind damit spielend leicht programmierbar.

Haben Sie schon einmal versucht, ein Programm möglichst benutzerfreundlich zu gestalten? Normalerweise geschieht dies über Menüs, die vom Anwender abgerufen werden und ihm die jeweilige aufgerufene Funktion erklären. Schön und gut. Nur, bis man sein Programm mit einem Menü versehen hat, vergeht meistens nochmal die gleiche Zeit wie bei der Entwicklung des gesamten Programms.



Das »Screen-Definition-Utility« läßt sich in Programme einbinden und erzeugt Bildschirmmasken mit allen Abfragen

Um Ihnen die Arbeit bei der Gestaltung von Menüs zu erleichtern, haben wir das »Screen-Definition-Utility« veröffentlicht, mit dem sich einfach und komfortabel eine komplette Bildschirmmaske aufbauen läßt. In diese lassen sich an beliebiger Stelle Abfragen nach Zahlen oder Zeichenketten einbauen. Anschließend schreibt unser Programm verschiedene Files auf Diskette, die in eigene Programme einbindbar sind und die die Bildschirmmaske mit allen Abfragen erzeugen.

»SDU« für meisterhafte Menüs

Das Programm ist in Turbo-Basic geschrieben (Listing 1). Es benötigt noch ein weiteres File. Dieses ist als AMPEL-Listing abgedruckt (Listing 2) und muß auf der gleichen Diskette wie das Screen-Definition-Utility unter dem Namen »SDU« vorhanden sein.

Nach dem Starten des Programms werden Sie nach dem Filenamen gefragt, unter dem die einzelnen von Screen-Definition-Utility erzeugten Files abgelegt werden. Geben Sie bitte keine Extension an, da das Programm mehrere Files mit dem angegebenen Namen erzeugt, zur Unterscheidung aber jeweils eine andere Extension einsetzt.

Anschließend fragt das Programm, ob das erzeugte Basic-Programm in kurzer oder langer Form gespeichert werden soll. Bei der langen Form wird aus jeder Zeichenfolge, die zwischen zwei Leerzeichen steht, ein eigener Print-Befehl erzeugt, die kurze Fassung verwandelt jede 40-Zeichen-Zeile vom Bildschirm in einen Print-Befehl.

Memory fragt nach der hexadezimalen Adresse, ab der Screen-Definition-Utility ein Maschinensprache-Programm erzeugt, das dann in eigenen Programme eingebunden wird. Dieses Maschinenprogramm lädt eine vorgefertigte Eingabemaske von der Diskette. Die Adresse muß zwischen $9000 und $CC40 liegen. Drücken Sie hier einfach nur <RETURN>, wird die Adresse $BC40 angenommen.

Schließlich folgt die Frage nach dem **Mark**-Zeichen. Mit diesem Zeichen markieren Sie in der Eingabemaske Stellen, an der Ihr späteres Programm eine Eingabe fordern soll. Dieses Zeichen ist normalerweise das <@>-Zeichen (Klammeraffe, Affenschaukel, wie auch immer), Sie können jedoch jedes beliebige andere Zeichen dafür definieren.

Beliebige Abfragepositionen

Nach dieser Eingabe gelangen Sie in den Editor, mit dem sich Eingabemasken konstruieren lassen. Falls auf der Diskette schon eine Maske unter dem im Eingangsmenü eingegebenen Namen vorhanden ist, wird diese automatisch geladen. Mit dem Cursor lassen sich an verschiedenen Stellen Zeichen setzen. Versehentlich gesetzte Zeichen löschen Sie wieder mit der <SPACE>-Taste. Eine ganze Zeile wird mit der Tastenkombination <SHIFT> und <DELETE> gelöscht, und folgende Zeilen werden nachgerückt. Die Bildschirmposition in der rechten untersten Ecke kann nicht belegt werden, da der Bildschirminhalt sonst eine Zeile nach oben rücken würde und damit der Bildschirmaufbau zerstört würde. Mit <ESC> verläßt man den Editor. Steht der Cursor beim Drücken von <ESC> in der obersten linken Ecke, werden die folgenden beiden Programmmteile nicht ausgeführt, und man befindet sich sofort im »Order«-Programm.

Steht der Cursor jedoch beim Verlassen des Editors an einer anderen Position, wird die Maske auf der Diskette gespeichert und anschließend die »Searching«-Routine aufgerufen. Diese durchsucht die Maske nach im Editor eingegebenen Mark-Zeichen. Findet die Search-Routine ein Mark, so fragt sie nach dem Variablennamen, den sie der an dieser Stelle eingebenen Zeichenfolge oder Zahl geben soll.

Sind alle Marks gefunden und der ganze Bildschirm durchsucht, wird die **Order**-Routine aufgerufen. Mit dieser legt man die Reihenfolge fest, in der die einzelnen Abfragen auf dem Bildschirm erscheinen sollen. Drücken Sie gleich zu Anfang <ESC> und wurde vorher die Searching-Routine nicht aufgerufen, so wird die Einstellung von einer vorher vorhandenen Maske übernommen und gleich der nächste Programmteil angesprungen. Ansonsten steht der Cursor auf der ersten Eingabe. Wollen Sie, daß diese Marke nicht als erstes in Ihrem Programm

abgefragt wird, springen Sie mit <+> oder <*> die nächste Marke an. Mit <RETURN> wird dann die Marke als nächstes in der Abfragereihenfolge festgelegt. Mit <ESC> löschen Sie eine komplette Abfragereihenfolge.

Ist die komplette Abfragereihenfolge der Marken gespeichert, so springt das Programm schließlich in das **Generate**-Menü. Hier werden Sie nach der ersten Zeilennummer gefragt, unter der die Eingabemaske in Ihrem Programm stehen soll, sowie nach dem Abstand der Zeilennummern. Anschließend wird das Eingabemasken-Programm auf Diskette gespeichert.

Auf Ihrer Diskette stehen dann insgesamt sechs Files, die das Eingabemaskenprogramm benötigt. Um die Maske in Ihr Programm einzubinden, geben Sie die Befehle »ENTER "D:Filename.BAS"« sowie ENTER "D:Filename.INP"« ein. In Ihrem Programm muß dann am Anfang noch der Befehl »BLOAD "D:Filename.ASM"« vorkommen. Jetzt wird die definierte Eingabemaske ohne Probleme geladen und ausgeführt.

(Daniel Sebastianelli/hf)

Steckbrief	
Programm:	Screen-Definition-Utility
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsumme und AMPEL
Datenträger:	Diskette

```
10 POKE 106,191                              <KO>
20 DIM PIC_NAME$(15),BAS_NAME$(15),MEM_N
AME$(15),EXT_NAME$(15),INP_NAME$(15),ASM
_NAME$(15),NAME$(11)                         <WG>
30 DIM Z(88,43),A$(40),KEY$(41),KEY1$(12
)                                            <ZA>
40 KEY$="$(,)0123456789ABCDEFGHIJKLMNOPQ
RSTUVWXYZ_"                                  <LR>
50 KEY1$="{ESC DEL}{ESC CTL <}{ESC CTL >
}{ESC SHIFT >}{ESC CTL DEL}{ESC SHIFT DE
L}{ESC CTL +}{ESC CTL *}{ESC CTL -}{ESC
CTL =}{ESC ESC}▲"                            <JQ>
60 POKE 82,0                                 <GC>
70 FOR I=0 TO 8                              <GK>
80 READ A$                                   <KY>
90 FOR J=0 TO 19                             <LV>
100 POKE $0600+J+I*20,ASC(A$(J+1))           <BH>
110 NEXT J                                   <GD>
120 NEXT I                                   <FV>
130 GRAPHICS 0                               <SM>
140 DL=DPEEK(560)                            <AK>
150 POKE DL-2,112                            <JL>
160 POKE DL-1,70                             <FE>
170 POKE DL+2,112                            <IR>
180 DPOKE 560,DL-2                           <JV>
190 BILD=DPEEK(88)                           <IB>
200 # MENUE                                  <FB>
210 CLS                                      <AU>
220 POKE 752,0                               <IW>
230 POKE 729,40                              <OT>
240 POKE 730,5                               <JY>
250 DPOKE DL,$0600+160                       <SR>
260 CLOSE #1                                 <LK>
270 POSITION 16,18                           <DB>
280 PUT 158                                  <PF>
290 I=LEN(NAME$)+1                           <LR>
300 TRAP #ERROR                              <DT>
310 OPEN #1,4,0,"D:SDU"                      <WY>
320 BGET #1,BILD,959                         <WE>
330 CLOSE #1                                 <LF>
340 PUT 158                                  <OY>
350 NAME$="D:"                               <JM>
360 I=2                                      <GE>
370 REPEAT                                   <DM>
380 I=I+1                                    <QS>
390 REPEAT                                   <DQ>
400 # INPUT_NAME                             <KP>
410 POKE 702,64                              <NZ>
420 POKE 694,0                               <LE>
430 TRAP #INPUT_NAME:GET KEY                 <JV>
440 IF I=3                                   <OM>
450 OK=INSTR(KEY$,CHR$(KEY),14)              <ZS>
460 ELSE                                     <YC>
470 OK=INSTR(KEY$,CHR$(KEY),4)               <TQ>
480 ENDIF                                    <VJ>
490 IF KEY=126 AND I>3                       <WR>
500 I=I-1                                    <QU>
510 PUT KEY                                  <RX>
520 ENDIF                                    <UY>
530 UNTIL OK>0 OR KEY=155 AND I>3            <BE>
540 PUT KEY                                  <SD>
550 IF KEY<>155 THEN NAME$(I)=CHR$(KEY)      <PG>
560 UNTIL I=10 OR KEY=155                    <HO>
570 NAME$=NAME$(1,I-1)                       <NK>
580 POSITION 16,20                           <YJ>
590 PUT 78                                   <UY>
600 REPEAT                                   <DB>
610 # INPUT_MIN_CODE                         <YK>
620 POKE 702,64                              <OD>
630 POKE 694,0                               <LI>
640 TRAP #INPUT_MIN_CODE:GET MIN_CODE        <GZ>
650 UNTIL MIN_CODE=74 OR MIN_CODE=78 OR
MIN_CODE=155                                 <XI>
660 IF MIN_CODE=155 THEN MIN_CODE=78         <KP>
670 POSITION 16,20                           <YI>
680 PUT MIN_CODE                             <JQ>
690 MEMORY=48192                             <LE>
700 A$="BC40"                                <RT>
710 POSITION 33,18                           <CD>
720 ? A$                                     <MN>
730 POSITION 33,18                           <CH>
740 PUT 158                                  <PC>
750 I=0                                      <FS>
760 REPEAT                                   <DO>
770 I=I+1                                    <QU>
780 REPEAT                                   <DS>
790 # INPUT_MEMORY                           <EL>
800 POKE 702,64                              <OB>
810 POKE 694,0                               <LG>
820 TRAP #INPUT_MEMORY:GET KEY               <VI>
830 HEX=INSTR(KEY$(5,20),CHR$(KEY))          <CH>
840 IF KEY=126 AND I>1                       <UT>
850 I=I-1                                    <RH>
860 PUT KEY                                  <SK>
870 ELSE                                     <YI>
880 IF I=1                                   <OE>
890 ? "▲▲▲▲"                                 <ED>
900 POSITION 33,18                           <CD>
910 PUT 158                                  <OY>
920 ENDIF                                    <VC>
930 ENDIF                                    <VE>
940 UNTIL HEX>0 OR KEY=155 AND I=1           <FR>
950 PUT KEY                                  <SJ>
960 A$(I)=CHR$(KEY)                          <UK>
970 UNTIL KEY=155 AND I=1 OR I=4             <OB>
980 IF I=4                                   <PJ>
990 MEMORY=DEC(A$(1,4))                      <ID>
1000 MEMORY=(MEMORY+36864+ABS(MEMORY-368
64))/2                                       <SL>
1010 MEMORY=(MEMORY+52288-ABS(MEMORY-522
88))/2                                       <QM>
1020 ENDIF                                   <SM>
1030 POSITION 33,18                          <TW>
1040 ? HEX$(MEMORY)                          <HT>
1050 MARK=64                                 <AA>
1060 POSITION 33,20                          <PB>
1070 PUT MARK                                <CF>
1080 REPEAT                                  <EX>
1090 REPEAT                                  <FA>
1100 # INPUT_MARK                            <GJ>
1110 POKE 702,64                             <VD>
1120 POKE 694,0                              <QN>
```

```
1130 TRAP #INPUT_MARK:GET KEY                  <OG>
1140 OK=INSTR(KEY$(5,40),CHR$(KEY))            <XP>
1150 OK1=INSTR(KEY1$,CHR$(KEY))                <IV>
1160 UNTIL OK=0 AND OK1=0 OR KEY=155           <AQ>
1170 IF KEY<>155 THEN MARK=KEY                 <UC>
1180 POSITION 33,20                            <PJ>
1190 PUT MARK                                  <CN>
1200 UNTIL KEY=155                             <XB>
1210 MARK=PEEK(BILD+833)                       <RP>
1220 POKE 729,10                               <UK>
1230 POKE 730,1                                <MY>
1240 PUT 158                                   <OR>
1250 CLS                                       <TY>
1260 I=LEN(NAME$)+1                            <BH>
1270 MEM_NAME$=NAME$                           <CP>
1280 MEM_NAME$(I)=".MEM"                       <GY>
1290 PIC_NAME$=NAME$                           <BN>
1300 PIC_NAME$(I)=".PIC"                       <BT>
1310 BAS_NAME$=NAME$                           <AF>
1320 BAS_NAME$(I)=".BAS"                       <XP>
1330 EXT_NAME$=NAME$                           <HQ>
1340 EXT_NAME$(I)=".EXT"                       <BU>
1350 INP_NAME$=NAME$                           <EW>
1360 INP_NAME$(I)=".INP"                       <QG>
1370 ASM_NAME$=NAME$                           <DR>
1380 ASM_NAME$(I)=".ASM"                       <KE>
1390 TRAP #EMPTY                               <UW>
1400 OPEN #1,4,0,PIC_NAME$                     <YI>
1410 BGET #1,BILD,959                          <PH>
1420 POKE 93,PEEK(BILD)                        <FN>
1430 # EMPTY                                   <PH>
1440 CLOSE #1                                  <ND>
1450 PUT 158                                   <OY>
1460 DPOKE DL,$0600                            <KG>
1470 REPEAT                                    <FC>
1480 REPEAT                                    <FF>
1490 # INPUT_CHAR                              <AK>
1500 TRAP #INPUT_CHAR:GET KEY                  <EZ>
1510 UNTIL (PEEK(84)<>23 OR KEY<>155) AN
D KEY<>125                                     <XH>
1520 IF PEEK(84)=23 AND PEEK(85)=39 THEN
 PUT 31                                        <WC>
1530 POKE 99,119                               <AO>
1540 PUT KEY                                   <UX>
1550 UNTIL KEY=27                              <PC>
1560 IF PEEK(84)=0 AND PEEK(85)=0              <GS>
1570 TRAP #MENUE                               <HR>
1580 OPEN #1,4,0,MEM_NAME$                     <CA>
1590 GET #1;X                                  <VK>
1600 FOR I=0 TO X                              <UY>
1610 FOR J=0 TO 2                              <WP>
1620 GET #1;Z                                  <VR>
1630 IF J=1                                    <KV>
1640 GET #1;ZH                                 <IR>
1650 Z=ZH*256+Z                                <TJ>
1660 ENDIF                                     <TK>
1670 Z(I,J)=Z                                  <GD>
1680 NEXT J                                    <GK>
1690 J=J-1                                     <MW>
1700 REPEAT                                    <EN>
1710 J=J+1                                     <LI>
1720 GET #1;Z                                  <VT>
1730 Z(I,J)=Z                                  <FT>
1740 UNTIL Z=0                                 <YB>
1750 NEXT I                                    <FS>
1760 MFLAG=1                                   <CD>
1770 CLOSE #1                                  <NS>
1780 TRAP #ERROR                               <RL>
1790 OPEN #1,4,0,PIC_NAME$                     <ZP>
1800 BGET #1,BILD,959                          <PM>
1810 CLOSE #1                                  <NC>
1820 ELSE                                      <TQ>
1830 POKE 752,1:PUT 27                         <XC>
1840 TRAP #ERROR                               <RB>
1850 OPEN #1,8,0,PIC_NAME$                     <BJ>
1860 BPUT #1,BILD,959                          <XN>
1870 CLOSE #1                                  <NU>
1880 DPOKE DL,$0600+40                         <WM>
1890 IM_BILD=BILD-1                            <FY>
1900 X=-1                                      <OX>
```

```
1910 Y=1                                       <ZN>
1920 REPEAT                                    <EX>
1930 IM_BILD=IM_BILD+1                         <HK>
1940 CHR=PEEK(IM_BILD)                         <GP>
1950 INV_CHR=(CHR+128) MOD 256                 <BQ>
1960 POKE IM_BILD,INV_CHR                      <WC>
1970 IF CHR<>91 AND (CHR>0 OR MIN_CODE=7
4)                                             <QO>
1980 IF CHR=MARK                               <ON>
1990 DPOKE DL,$0600+20                         <VB>
2000 CHR=0                                     <UJ>
2010 I=5                                       <WK>
2020 Y=1                                       <YZ>
2030 X=X+1                                     <SP>
2040 Z(X,Y)=IM_BILD-BILD                       <BW>
2050 Y=Y+1                                     <TJ>
2060 Z(X,Y)=0                                  <UZ>
2070 Y=Y+1                                     <TP>
2080 REPEAT                                    <EY>
2090 REPEAT                                    <FB>
2100 # INPUT_VAR                               <OF>
2110 POKE 702,64                               <VE>
2120 POKE 694,0                                <QO>
2130 TRAP #INPUT_VAR:GET KEY                   <QX>
2140 IF I=5                                    <LY>
2150 OK=INSTR(KEY$,CHR$(KEY),14)               <WW>
2160 ELSE                                      <TP>
2170 OK=INSTR(KEY$,CHR$(KEY))                  <GT>
2180 ENDIF                                     <TH>
2190 IF KEY=126 AND I>5                        <XB>
2200 I=I-1                                     <LA>
2210 POKE $0600+20+I,0                         <EA>
2220 ENDIF                                     <SR>
2230 UNTIL OK>0 OR KEY=155                     <GD>
2240 IF KEY<>126 AND KEY<>155                  <OR>
2250 POKE $0600+20+I,KEY-32                    <ES>
2260 I=I+1                                     <LA>
2270 ENDIF                                     <TG>
2280 UNTIL KEY=155 OR I=19                     <KM>
2290 DPOKE DL,$0600+40                         <WE>
2300 FOR J=5 TO I-1                            <CP>
2310 Z(X,Y)=PEEK($0600+20+J)+32                <VP>
2320 Y=Y+1                                     <TG>
2330 POKE $0600+20+J,0                         <FC>
2340 NEXT J                                    <FT>
2350 Y=1                                       <ZO>
2360 ELSE                                      <TT>
2370 IF Y=1                                    <QH>
2380 X=X+1                                     <TK>
2390 Z(X,Y)=IM_BILD-BILD                       <CR>
2400 Y=Y+1                                     <TC>
2410 Z(X,Y)=1                                  <VF>
2420 Y=Y+1                                     <TI>
2430 ENDIF                                     <SY>
2440 IF CHR=64 THEN CHR=0                      <WK>
2450 IF CHR<64                                 <FK>
2460 ASC_CHR=CHR+32                            <BE>
2470 ELSE                                      <TY>
2480 IF CHR>63 AND CHR<96                      <CT>
2490 ASC_CHR=CHR-64                            <GJ>
2500 ELSE                                      <TF>
2510 IF CHR>127 AND CHR<192                    <ZP>
2520 ASC_CHR=CHR+32                            <AU>
2530 ELSE                                      <TO>
2540 IF CHR>191 AND CHR<224                    <WH>
2550 ASC_CHR=CHR-64                            <FZ>
2560 ELSE                                      <TX>
2570 ASC_CHR=CHR                               <FK>
2580 ENDIF                                     <TP>
2590 ENDIF                                     <TS>
2600 ENDIF                                     <ST>
2610 ENDIF                                     <SW>
2620 Z(X,Y)=ASC_CHR                            <MW>
2630 Y=Y+1                                     <TP>
2640 IF (IM_BILD-BILD+1) MOD 40=0 THEN Y
=1                                             <FZ>
```

Listing 1. »Screen-Definition-Utility«. Bitte mit Prüfsummer eingeben.

```
2650 ENDIF                                    <TI>
2660 ELSE                                     <TZ>
2670 Y=1                                      <AA>
2680 ENDIF                                    <TR>
2690 POKE IM_BILD,CHR                         <LB>
2700 IF X>87                                  <SE>
2710 GRAPHICS 0                               <BL>
2720 PUT 25                                   <PK>
2730 ? "ERROR-▲MEHR▲ALS▲88▲FELDER"            <JU>
2740 ? "VERSUCHE▲MINIMUM-CODE=""J"""          <BR>
2750 END                                      <QT>
2760 ENDIF                                    <TN>
2770 UNTIL CHR=91                             <JY>
2780 IF X=-1 THEN GO# MENUE                   <HD>
2790 POKE IM_BILD,0                           <FQ>
2800 TRAP #ERROR                              <QQ>
2810 OPEN #1,8,0,EXT_NAME$                    <RJ>
2820 BPUT #1,BILD,959                         <XC>
2830 CLOSE #1                                 <HJ>
2840 TRAP #ERROR                              <RC>
2850 OPEN #1,8,0,ASM_NAME$                    <GA>
2860 PUT #1,255                               <IX>
2870 PUT #1,255                               <JA>
2880 PUT #1,MEMORY MOD 256                    <UI>
2890 PUT #1,MEMORY DIV 256                    <XX>
2900 PUT #1,(MEMORY+958) MOD 256              <HS>
2910 PUT #1,(MEMORY+958) DIV 256              <LZ>
2920 BPUT #1,BILD,959                         <XE>
2930 CLOSE #1                                 <HL>
2940 ENDIF                                    <TL>
2950 DPOKE DL,$0600+60                        <XH>
2960 REPEAT                                   <FK>
2970 POS=Z(Z,1)+BILD                          <LS>
2980 IF Z(Z,2)=1                              <DF>
2990 CHR=PEEK(POS)                            <TX>
3000 ELSE                                     <SH>
3010 CHR=31                                   <QK>
3020 ENDIF                                    <SO>
3030 INV_CHR=(CHR+128) MOD 256                <AU>
3040 POKE POS,INV_CHR                         <SK>
3050 TRAP #ERROR                              <QQ>
3060 GET KEY                                  <OL>
3070 IF KEY=27                                <KA>
3080 IF MFLAG=1                               <XO>
3090 EXIT                                     <ZG>
3100 ELSE                                     <SY>
3110 FOR I=0 TO X                             <UT>
3120 Z(I,0)=0                                 <AG>
3130 NEXT I                                   <FC>
3140 L=0                                      <HA>
3150 ENDIF                                    <SZ>
3160 ELSE                                     <TQ>
3170 IF MFLAG=1                               <XN>
3180 FOR I=0 TO X                             <VO>
3190 Z(I,0)=0                                 <BB>
3200 NEXT I                                   <EV>
3210 L=0                                      <VT>
3220 MFLAG=0                                  <AX>
3230 ENDIF                                    <SV>
3240 IF KEY=42                                <ID>
3250 # RIGHT                                  <HF>
3260 REPEAT                                   <EX>
3270 Z=Z+1                                    <UI>
3280 IF Z>X THEN Z=0                          <RR>
3290 UNTIL Z(Z,0)=0 OR L-1=X                  <AD>
3300 ELSE                                     <TC>
3310 IF KEY=43                                <IK>
3320 REPEAT                                   <EN>
3330 Z=Z-1                                    <UQ>
3340 IF Z<0 THEN Z=X                          <EJ>
3350 UNTIL Z(Z,0)=0                           <NC>
3360 ELSE                                     <TU>
3370 IF KEY=155                               <NG>
3380 L=L+1                                    <HZ>
3390 Z(Z,0)=L                                 <UL>
3400 GO# RIGHT                                <AZ>
3410 ENDIF                                    <ST>
3420 ENDIF                                    <SH>
3430 ENDIF                                    <SZ>
3440 ENDIF                                    <TC>
```

```
3450 POKE POS,CHR                             <GS>
3460 UNTIL L-1=X                              <VO>
3470 PUT 158                                  <PG>
3480 CLS                                      <UN>
3490 POKE 752,0:PUT 158                       <SA>
3500 DPOKE DL,$0600+100                       <JL>
3510 POSITION 13,8                            <IH>
3520 ? "Line▲Number:";                        <HS>
3530 POSITION 16,10                           <PE>
3540 ? "Increase:";                           <LC>
3550 REPEAT                                   <FA>
3560 POSITION 26,8                            <LH>
3570 ? "10",                                  <ZR>
3580 POSITION 26,10                           <QI>
3590 ? "10",                                  <ZX>
3600 REPEAT                                   <EN>
3610 # INPUT_LINE                             <EQ>
3620 POSITION 26,8                            <LM>
3630 TRAP #INPUT_LINE:INPUT "";LINE           <BZ>
3640 LINE=ABS(TRUNC(LINE))                    <VS>
3650 UNTIL LINE>=0 AND LINE<32766             <QY>
3660 POSITION 26,8                            <LY>
3670 ? LINE,                                  <VI>
3680 REPEAT                                   <FL>
3690 # INPUT_INC                              <GY>
3700 POSITION 26,10                           <PO>
3710 TRAP #INPUT_INC:INPUT "";INC             <KM>
3720 INC=ABS(TRUNC(INC))                      <JT>
3730 UNTIL INC>0                              <HF>
3740 UNTIL LINE+((X+1)*2+1)*INC<32767         <BK>
3750 POSITION 26,10                           <QD>
3760 ? INC,                                   <MZ>
3770 Z=0                                      <AB>
3780 TRAP #ERROR                              <RN>
3790 OPEN #1,8,0,BAS_NAME$                    <YP>
3800 OPEN #2,8,0,INP_NAME$                    <KD>
3810 ? #1;LINE;"▲GRAPHICS▲0"                  <QG>
3820 ? #2;LINE;"▲GRAPHICS▲0"                  <QS>
3830 LINE=LINE+INC                            <BJ>
3840 ? #2;LINE;"▲OPEN▲#1,4,0,""";EXT_NAM
E$;""""                                       <OC>
3850 LINE=LINE+INC                            <BP>
3860 ? #2;LINE;"▲BGET▲#1,DPEEK(88),959"       <HA>
3870 LINE=LINE+INC                            <BV>
3880 ? #2;LINE;"▲POKE▲93,PEEK(DPEEK(88))
"                                             <HS>
3890 LINE=LINE+INC                            <CB>
3900 I=-1                                     <LN>
3910 X=X+1                                    <TC>
3920 REPEAT                                   <EZ>
3930 Z=Z+1                                    <UK>
3940 REPEAT                                   <FF>
3950 I=I+1                                    <LM>
3960 UNTIL Z(I,0)=Z                           <JF>
3970 DX=Z(I,1) MOD 40                         <BL>
3980 DY=Z(I,1) DIV 40                         <EV>
3990 ? #1;LINE;"▲POSITION▲";DX;",";DY;":
POKE▲99,119"                                  <SI>
4000 IF Z(I,2)=0 THEN ? #2;LINE;"▲POSITI
ON▲";DX;",";DY;":POKE▲99,119"                 <FD>
4010 LINE=LINE+INC                            <AO>
4020 J=3                                      <HF>
4030 IF Z(I,2)=1                              <UV>
4040 ? #1;LINE;"▲?▲""";                       <YH>
4050 LINE=LINE+INC                            <BA>
4060 REPEAT                                   <EU>
4070 ? #1;CHR$(Z(I,J));                       <HJ>
4080 IF Z(I,J)=34 THEN ? #1;CHR$(34);         <CA>
4090 J=J+1                                    <LV>
4100 UNTIL Z(I,J)=0                           <TG>
4110 ? #1;""";"                               <PA>
4120 ELSE                                     <TF>
4130 A$="DUMMY:"                              <GX>
4140 ? #1;LINE;"▲TRAP▲";LINE-INC;":INPUT
▲"""";";                                      <IH>
4150 ? #2;LINE;"▲TRAP▲";LINE-INC;":INPUT
▲"""";";                                      <IT>
4160 LINE=LINE+INC                            <BF>
4170 REPEAT                                   <EZ>
4180 FLAG=0                                   <UT>
```

```
4190 IF Z(I,J)=40                              <OV>
4200 IF Z(I,J-1)=36 THEN A$(6,7)="$:"          <II>
4210 ? #1;A$;                                  <QG>
4220 ? #2;A$;                                  <QS>
4230 J=3                                       <WM>
4240 REPEAT                                    <ES>
4250 ? #1;CHR$(Z(I,J));                        <NH>
4260 ? #2;CHR$(Z(I,J));                        <NT>
4270 J=J+1                                     <LT>
4280 UNTIL Z(I,J)=0                            <UG>
4290 ? #1;"=";A$(1,LEN(A$)-1)                  <HE>
4300 ? #2;"=";A$(1,LEN(A$)-1)                  <GO>
4310 FLAG=1                                    <UN>
4320 ENDIF                                     <SV>
4330 J=J+1                                     <LJ>
4340 UNTIL Z(I,J)=0                            <TW>
4350 IF FLAG=0                                 <HF>
4360 J=3                                       <WX>
4370 REPEAT                                    <FD>
4380 ? #1;CHR$(Z(I,J));                        <NS>
4390 ? #2;CHR$(Z(I,J));                        <OE>
4400 J=J+1                                     <LC>
4410 UNTIL Z(I,J)=0                            <TP>
4420 ? #1;CHR$(155);                           <EU>
4430 ? #2;CHR$(155);                           <FG>
4440 ENDIF                                     <TD>
4450 ENDIF                                     <TG>
4460 I=-1                                      <LW>
4470 UNTIL Z=X                                 <TV>
4480 TRAP #ERROR                               <RI>
4490 CLOSE #1                                  <NV>
4500 OPEN #1,8,0,MEM_NAME$                     <DJ>
4510 X=X-1                                     <TN>
4520 PUT #1;X                                  <BC>
4530 FOR I=0 TO X                              <VI>
4540 FOR J=0 TO 2                              <WZ>
4550 Z=Z(I,J)                                  <XV>
4560 PUT #1;Z                                  <CO>
4570 IF J=1 THEN PUT #1;Z DIV 256              <AZ>
4580 NEXT J                                    <GL>
4590 J=J-1                                     <MX>
4600 REPEAT                                    <EO>
4610 J=J+1                                     <LJ>
4620 PUT #1;Z(I,J)                             <NR>
4630 UNTIL Z(I,J)=0                            <TZ>
4640 NEXT I                                    <FQ>
4650 CLOSE #1                                  <NN>
4660 DPOKE DL,$0600+120                        <LY>
4670 ?                                         <XF>
4680 ?                                         <XI>
4690 ? "NEW"                                   <JY>
4700 ?                                         <WM>
4710 ?                                         <WP>
4720 ? "ENTER▲""";BAS_NAME$                    <NJ>
4730 ? "ENTER▲""";INP_NAME$                    <AE>
4740 ? "BLOAD▲""";ASM_NAME$                    <JC>
4750 END                                       <QV>
4760 # ERROR                                   <ND>
4770 CLOSE                                     <ZE>
4780 DL_OLD=DPEEK(DL)                          <KM>
4790 DPOKE DL,$0600+140                        <OB>
4800 POKE 764,255                              <GK>
4810 GET KEY                                   <ON>
4820 DPOKE DL,DL_OLD                           <KY>
4830 TRAP #ERROR                               <RB>
4840 GOTO ERL                                  <AB>
4850 DATA▲@@@@@@@editor@@@@@@@                 <MN>
4860 DATA▲nameZ@@@@@@@@@@@@@@@@                <ZH>
4870 DATA▲@@@@@@searching@@@@@                 <ZE>
4880 DATA▲@@@@@@@@@order@@@@@@@                <GV>
4890 DATA▲nameZ@@@@@@@@@@@@@@@@                <ZQ>
4900 DATA▲@@@@@@generate@@@@@@                 <MF>
4910 DATA▲@@@@@@@@ready@@@@@@@                 <BF>
4920 DATA▲@disc@ok_@press@key@                 <QS>
4930 DATA▲@@@@@@@@menu@@@@@@@@                 <YG>
```

Listing 1 (Schluß)

```
Programmname :SDU
Länge        :960 Bytes
--------------------------------

0000:51 52 52 52 52 52 52 52 <5B>
0008:52 52 52 52 52 52 52 52 <B3>
0010:52 52 52 52 52 52 52 52 <CB>
0018:52 52 52 52 52 52 52 52 <F3>
0020:52 52 52 52 52 52 52 45 <EE>
0028:7C 00 00 B3 80 A3 80 B2 <D3>
0030:80 A5 80 A5 80 AE 80 80 <0F>
0038:80 80 A4 80 A5 80 A6 80 <B7>
0040:A9 80 AE 80 A9 80 B4 80 <95>
0048:A9 80 AF 80 AE 00 00 7C <B4>
0050:7C 00 00 00 00 00 00 00 <0E>
0058:00 00 00 00 00 00 B5 80 <43>
0060:B4 80 A9 80 AC 80 A9 80 <59>
0068:B4 80 B9 00 00 00 00 00 <99>
0070:00 00 00 00 00 00 00 7C <EC>
0078:41 52 52 52 52 52 52 52 <6B>
0080:52 57 52 52 52 52 52 52 <99>
0088:52 52 52 52 52 52 52 52 <3F>
0090:52 52 52 52 52 52 57 52 <51>
0098:52 52 52 52 52 52 52 44 <60>
00A0:7C 00 00 00 00 00 00 00 <DE>
00A8:00 7C 00 00 00 00 30 72 <59>
00B0:6F 67 72 61 6D 6D 69 65 <B5>
00B8:72 74 00 00 00 00 7C 00 <C7>
00C0:00 00 00 00 00 00 00 7C <3C>
00C8:7C 00 00 00 00 00 00 00 <86>
00D0:00 7C 00 00 00 00 00 00 <EF>
00D8:00 00 00 76 6F 6E 00 00 <61>
00E0:00 00 00 00 00 00 7C 00 <D9>
00E8:00 00 00 00 00 00 00 7C <64>
00F0:7C 00 00 00 00 00 00 00 <AE>
00F8:00 7C 24 61 6E 69 65 6C <B0>
0100:00 33 65 62 61 73 74 69 <BC>
0108:61 6E 65 6C 6C 69 7C 00 <6B>
0110:00 00 00 00 00 00 00 7C <8E>
0118:7C 00 00 00 00 00 00 00 <58>
0120:00 7C 00 00 00 00 00 00 <01>
0128:00 00 10 12 0F 18 17 00 <54>
0130:00 00 00 00 00 00 7C 00 <2B>
0138:00 00 00 00 00 00 00 7C <B6>
0140:7C 00 00 00 00 00 00 00 <00>
0148:00 5A 52 52 52 52 52 52 <0E>
0150:52 52 52 52 52 52 52 52 <09>
0158:52 52 52 52 52 52 43 00 <41>
0160:00 00 00 00 00 00 00 7C <DE>
0168:7C 00 00 00 00 00 00 00 <28>
0170:00 00 00 00 00 00 00 00 <72>
0178:00 00 00 00 00 00 00 00 <7A>
0180:00 00 00 00 00 00 00 00 <82>
0188:00 00 00 00 00 00 00 7C <06>
0190:7C 00 2B 75 72 7A 62 65 <2C>
0198:73 63 68 72 65 69 62 75 <5B>
01A0:6E 67 1A 00 00 00 00 00 <F6>
01A8:00 00 00 00 00 00 00 00 <AA>
01B0:00 00 00 00 00 00 00 7C <2E>
01B8:7C 00 00 00 00 00 00 00 <F8>
01C0:00 00 00 00 00 00 00 00 <C2>
01C8:00 00 00 00 00 00 00 00 <CA>
01D0:00 00 00 00 00 00 00 00 <D2>
01D8:00 00 00 00 00 00 00 7C <56>
01E0:7C 00 00 2D 69 74 00 28 <B4>
01E8:69 6C 66 65 00 64 69 65 <06>
01F0:73 65 73 00 30 72 6F 67 <44>
01F8:72 61 6D 6D 73 00 6B 6F <1F>
0200:65 6E 6E 65 6E 00 00 7C <0D>
0208:7C 00 00 33 69 65 00 6D <C3>
0210:69 74 00 65 69 6E 65 6D <73>
0218:00 26 75 6C 6C 0D 33 63 <7C>
0220:72 65 65 6E 0D 25 64 69 <58>
0228:74 6F 72 00 00 00 00 7C <CC>
0230:7C 00 00 22 21 33 29 23 <DF>
0238:0D 23 6F 64 65 00 66 75 <24>
0240:65 72 00 22 69 6C 64 73 <A6>
0248:63 68 69 72 6D 6D 61 73 <5E>
0250:6B 65 6E 00 00 00 00 7C <2C>
0258:7C 00 00 67 65 6E 65 72 <9C>
0260:69 65 72 65 6E 0E 00 00 <1A>
0268:00 00 00 00 00 00 00 00 <6C>
0270:00 00 00 00 00 00 00 00 <74>
0278:00 00 00 00 00 00 00 7C <F8>
0280:7C 00 00 00 00 00 00 00 <C2>
0288:00 00 00 00 00 00 00 00 <8C>
0290:00 00 00 00 00 00 00 00 <94>
0298:00 00 00 00 00 00 00 00 <9C>
02A0:00 00 00 00 00 00 00 7C <20>
02A8:7C 00 00 00 00 00 00 00 <EA>
02B0:00 00 00 00 00 00 00 00 <B4>
02B8:00 00 00 00 00 00 00 00 <BC>
02C0:00 00 00 00 00 00 00 00 <C4>
02C8:00 00 00 00 00 00 00 7C <48>
02D0:7C 00 00 00 00 26 69 6C <69>
02D8:65 00 2E 61 6D 65 1A 00 <1E>
02E0:00 00 00 00 00 00 00 00 <E4>
02E8:00 2D 65 6D 6F 72 79 1A <E9>
02F0:00 22 23 14 10 00 00 7C <FF>
02F8:7C 00 00 00 00 00 00 00 <DA>
0300:00 00 00 00 00 00 00 00 <06>
0308:00 00 00 00 00 00 00 00 <0E>
0310:00 00 00 00 00 00 00 00 <16>
0318:00 00 00 00 00 00 00 7C <9A>
0320:7C 00 2D 69 6E 69 6D 75 <F7>
0328:6D 00 23 6F 64 65 1A 00 <19>
0330:2E 00 00 00 00 00 00 00 <4D>
0338:00 00 00 2D 61 72 6B 1A <C4>
0340:00 20 00 00 00 00 00 7C <CA>
0348:7C 00 00 00 00 00 00 00 <0C>
0350:00 00 00 00 00 00 00 00 <56>
0358:00 00 00 00 00 00 00 00 <5E>
0360:00 00 00 00 00 00 00 00 <66>
0368:00 00 00 00 00 00 00 7C <EA>
0370:5A 52 52 52 52 52 52 52 <D1>
0378:52 52 52 52 52 52 52 52 <DD>
0380:52 52 52 52 52 52 52 52 <35>
0388:52 52 52 52 52 52 36 65 <28>
0390:72 73 0E 12 0E 10 52 43 <23>
0398:00 00 00 00 00 00 00 00 <9E>
03A0:00 00 00 00 00 00 00 00 <A6>
03A8:00 00 00 00 00 00 00 00 <AE>
03B0:00 00 00 00 00 00 00 00 <B6>
03B8:00 00 00 00 00 00 00 00 <BE>
```

Listing 2. »SDU«. Bitte mit AMPEL eingeben.

Prüfsummer paßt auf

Unser Prüfsummer für alle Atari-Computer ist besonders komfortabel. Mit Hilfe dieses Programms lassen sich Listings auf Anhieb korrekt und problemlos eingeben.

Mag der Atari-Prüfsummer auf den ersten Blick etwas umfangreich erscheinen, der Aufwand lohnt sich. So legt der Prüfsummer zum Beispiel eine zusätzliche Statuszeile am oberen Bildschirmrand an. Diese Zeile ist von Basic aus nicht zu erreichen, also auch nicht zu löschen. Weiterhin können Sie den Prüfsummer auch für die Eingabe von Turbo-Basic-XL-Programmen verwenden. Beachten Sie hierzu die besonderen Hinweise!

In dem abgedruckten Listing finden Sie gewisse Zeichen vor, die der Atari nicht kennt. Es handelt sich dabei einerseits um Dreiecke und andererseits um geschweifte Klammern. Die Dreiecke stellen grundsätzlich Leerstellen dar; Texte, die zwischen geschweiften Klammern stehen, repräsentieren Atari-spezifische Grafik- und Sonderzeichen. Ein Beispiel: »ESC CTL=« entspricht dem ASCII-Zeichen 29. Um dieses Zeichen einzugeben, betätigen Sie zunächst die <ESC>-Taste und dann gleichzeitig die <CONTROL>- und die <=>-Taste. Daraufhin erscheint das gewünschte Zeichen auf dem Bildschirm. Eine komplette Aufstellung aller Grafik- und Sonderzeichen finden Sie in der Tabelle. Weiterhin gibt es unterstrichene Zeichen, die eine inverse Darstellung bedeuten. Bevor Sie ein solches Zeichen eingeben, müssen Sie unbedingt die Atari-Taste betätigen.

Bild 1 zeigt eine Programmzeile in der üblichen Darstellungsweise und zum Vergleich dazu eine in konvertierter Form.

```
4850 POSITION 11,2:? "L────────────────|":
POSITION 4,5:? "FILENAME      SIZE":POSITI
ON 2,7


4850 POSITION 11,2:? "{CTL Z}{CTL R}{CTL
 R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}{C
TL R}{CTL R}{CTL R}{CTL R}{CTL R}{CTL R}
{CTL R}{SHIFT =}":POSITION 4,5:? "FILENA
ME▲▲▲▲SIZE":POSITION 2,7
```

Bild 1. Oben eine Programmzeile, wie sie auf dem Bildschirm erscheint. Darunter dieselbe Zeile in der konvertierten Form.

Geben Sie also zunächst das Prüfsummenprogramm von Basic - oder Turbo-Basic XL - aus ein. Zwar werden sämtliche DATA-Werte nach dem Start mit RUN überprüft, aber »nobody is perfect«.

Kontrollieren Sie also das Prüfsummenprogramm vor dem Start noch einmal. Speichern Sie es dann auf einem Datenträger. Jetzt können Sie das Programm beruhigt mit RUN starten. Aus den DATA-Werten wird nun ein Maschinen-Programm erzeugt. Sollte sich beim Eintippen des Programms ein falscher DATA-Wert eingeschlichen haben, so wird die entsprechende Zeilennummer auf dem Bildschirm angezeigt.

Nach der Korrektur starten Sie das Programm erneut mit RUN. Wurde das Programm fehlerlos eingegeben, stehen folgende Funktionen zur Auswahl:

1. Das Prüfsummen-Programm als Maschinensprache-File auf Diskette schreiben. Nach Betätigung der Taste 1 fragt das Programm nach dem Namen für das Prüfsummenprogramm. Tippen Sie also beispielsweise »PRUEFER.OBJ« ein, so wird nach Druck auf die <RETURN>-Taste das Prüfsummenprogramm auf Diskette gespeichert. Der Computer schließt den Vorgang mit der »READY«-Meldung ab.

Um das Prüfcode-Programm verwenden zu können, löschen Sie den Programmspeicher zunächst mit NEW. Vom DOS-Menü aus (einfach DOS eintippen und die <RETURN>-Taste betätigen) wählen Sie die Option L und geben den Namen des vorher erzeugten Maschinenprogramms ein. Jetzt wird das Prüfsummenprogramm in den RAM-Speicher geladen. Mit <B> gelangen Sie wieder zurück ins Basic.

```
Pruefcode :  >  <  (c) 85 by W. Kress/wb
 1000 PROC RECHNEN
 1010    GRAPHICS 15+16:POKE 764,255
 1030    DX=(XMAX-XMIN)/159
 1040    DY=(YMAX-YMIN)/191
 1050    CX=XMIN:CY=YMAX
 1080    FOR ZEILE=0 TO 191
 1090      FOR SPALTE=0 TO 159
 1110        TI=0:XW=0:YW=0:X2=0:Y2=0
 1120        WHILE TI<TMAX AND (X2+Y2)<8

 1130          YW=2*XW*YW-CY:XW=X2-Y2-CX

 1150          X2=XW^2:Y2=YW^2:TI=TI+1
 1160        WEND
 1170        IF TI=TMAX
 1200          COLOR %0
 1220        ELSE
 1230          COLOR (TI MOD 3)+%1
 1240        ENDIF
 1250        CX=CX+DX:PLOT SPALTE,ZEILE
 1260        IF PEEK(764)=28 THEN POP :G
OTO 1295
 1270      NEXT SPALTE
 1280      C
```

Bild 2. So sollte der Bildschirm nach dem Aktivieren des Prüfsummers aussehen

Aktiviert wird jetzt das Prüfsummenprogramm mit »?USR(1536)«. Am oberen Bildschirmrand erhalten Sie daraufhin eine zusätzliche Zeile mit dem Schriftzug »Prüfcode: > < (c) by W. Kress/wb«. Wie die Aufteilung des Bildschirms unter Verwendung des Prüfsummenprogramms aussieht, zeigt Bild 2.

2. Das Maschinenprogramm wird als Boot-File auf Kassette geschrieben. Das spätere Laden oder Booten geht folgendermaßen vor sich:
- bei ausgeschaltetem Computer die <START>-Taste drücken und Computer einschalten,
- nach Ertönen des Summtons die <START>-Taste loslassen,
- nacheinander die <PLAY>-Taste des Kassettenrecorders und die <RETURN>-Taste betätigen,
- das Prüfsummenprogramm wird geladen und automatisch gestartet

3. Programm beenden. Das Maschinenprogramm wird nicht gespeichert.

Das Prüfsummenprogramm wird in der Page 6 abgelegt, einem vor NEW geschützten Speicherbereich. Abschalten kann man es durch Drücken von <SYSTEM RESET>, ein Neustart erfolgt mit dem Aufruf: »?USR(1536)«.

Sobald das Prüfsummenprogramm aktiviert ist, können beim Zugriff auf ein Speichermedium Bildstörungen auftreten. Um dies zu vermeiden, brauchen Sie nach dem Laden oder nach dem Aufruf mit »?USR(1536)« nur »POKE 39998,30« einzugeben.

Wurde das Programm von Kassette gebootet, wird es jedesmal nach Betätigung der <SYSTEM RESET>-Taste

erneut gestartet. Gibt man jedoch vor der Betätigung der <SYSTEM RESET>-Taste »POKE 9,0« ein, kann man diesen Effekt umgehen.

Bevor Sie nun ein eingetipptes Basic- oder Turbo-Basic XL-Programm mit RUN starten, sollten Sie es sicherheitshalber noch einmal speichern. Weiterhin sollte das Prüfsummen-Programm mit <SYSTEM RESET> desaktiviert werden, da auch manche Basic-Programme die Page 6 für Maschinen-Unterprogramme verwenden. Es könnte sonst zu Überschneidungen und zum Systemabsturz kommen.

Der Prüfcode besteht jeweils aus zwei Zeichen. Er setzt sich aus den Buchstaben von A bis Z zusammen. Geben Sie nun probeweise eine Programmzeile ein und schließen Sie mit der <RETURN>-Taste ab. Der zugehörige Prüfcode erscheint dann in der obersten Zeile. Sollte der Code nicht mit dem im abgedruckten Listing übereinstimmen, kann die Programmzeile sofort verbessert werden. Die jeweilige Programmzeile braucht dazu nicht erneut eingegeben zu werden, da sämtliche Editierfunktionen des Atari-Computers zur Verfügung stehen.

Ein wichtiger Hinweis:

Für die erste Programmzeile kann eventuell ein falscher Prüfcode errechnet werden. Sollte die Prüfsumme also nicht mit der im Listing abgedruckten übereinstimmen, fahren Sie mit dem Cursor nochmals auf die erste Zeile. Sobald Sie die <RETURN>-Taste betätigt haben, erscheint der richtige Code auf dem Bildschirm. Stimmt der ausgegebene Wert dann immer noch nicht, überprüfen Sie nochmals jedes einzelne Zeichen der von Ihnen eingegebenen Programmzeile.

Bei der Eingabe von Programmen sind die üblichen Abkürzungen für Basic-Befehle zulässig. Sie können also zum Beispiel G. für GOTO, SE. für SETCOLOR und so weiter verwenden. Eine Ausnahme gilt jedoch bei der Abkürzung für den PRINT-Befehl, den man in der Regel mit einem Fragezeichen abkürzen kann. **Verwenden Sie das Fragezeichen bitte nur dann, wenn es auch im Listing verwendet wird.**

Möchten Sie nachträglich ein Programm überprüfen, so laden Sie zuerst das Prüfsummenprogramm und starten es mit »?USR(1536)«. Anschließend laden Sie Ihr Programm und listen einen Teil auf dem Bildschirm. Dann bewegen Sie den Cursor an irgendeine Stelle einer Programmzeile. Mit Betätigung der <RETURN>-Taste wird der Prüfcode in der Statuszeile dargestellt. Stimmt der Prüfcode mit dem im beigefügten Listing überein, betätigen Sie wieder die <RETURN>-Taste und der Prüfcode der nächsten Zeile wird errechnet und angezeigt. Auf diese Art und Weise können Sie ein komplettes Programm oder auch nur einzelne Zeilen »durchchecken«.

Sollte der Code nicht mit dem des Listings übereinstimmen, überprüfen Sie Ihre Programmzeile. Denn bei allen abgetippten Listings gilt: Die Programme müssen so abge-

Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)	Dezimal-Code	ATASCII-Zeichen	zu betätigende Taste(n)
A	B	C	A	B	C	A	B	C	A	B	C
0		CTL .	19		CTL S	128		(Invers) CTL .	147		(Invers) CTL S
1		CTL A	20		CTL T	129		(Invers) CTL A	148		(Invers) CTL T
2		CTL B	21		CTL U	130		(Invers) CTL B	149		(Invers) CTL U
3		CTL C	22		CTL V	131		(Invers) CTL C	150		(Invers) CTL V
4		CTL D	23		CTL W	132		(Invers) CTL D	151		(Invers) CTL W
5		CTL E	24		CTL X	133		(Invers) CTL E	152		(Invers) CTL X
6		CTL F	25		CTL Y	134		(Invers) CTL F	153		(Invers) CTL Y
7		CTL G	26		CTL Z	135		(Invers) CTL G	154		(Invers) CTL Z
8		CTL H	27		ESC ESC	136		(Invers) CTL H	156		ESC SHIFT DEL
9		CTL I	28		ESC CTL –	137		(Invers) CTL I	157		ESC SHIFT >
10		CTL J	29		ESC CTL =	138		(Invers) CTL J	158		ESC CTL TAB
11		CTL K	30		ESC CTL +	139		(Invers) CTL K	159		ESC SHIFT TAB
12		CTL L	31		ESC CTL *	140		(Invers) CTL L	224		(Invers) CTL .
13		CTL M	96		CTL .	141		(Invers) CTL M	251		(Invers) CTL ;
14		CTL N	123		CTL ;	142		(Invers) CTL N	252		(Invers) SHIFT =
15		CTL O	124		SHIFT =	143		(Invers) CTL C	253		ESC CTL 2
16		CTL P	125		ESC CTL <	144		(Invers) CTL P	254		ESC CTL DEL
17		CTL Q	126		ESC DEL	145		(Invers) CTL Q	255		ESC CTL >
18		CTL R	127		ESC TAB	146		(Invers) CTL R			

Tabelle der Atari-Grafikzeichen

tippt werden, wie sie gedruckt sind. Stimmt die Anzahl der Leerzeichen? Haben Sie eventuell Zeichen vertauscht? Oder haben Sie ein »?« anstelle eines PRINT-Befehls eingegeben? Haben Sie den Fehler gefunden? Dann weiterhin viel Erfolg mit Ihren Atari-Programmen!

Besondere Hinweise zum Prüfsummenprogramm

Wenn Sie ein Turbo-Basic XL-Programm eingeben möchten, müssen Sie das Prüfsummenprogramm mit einem Hilfsprogramm starten. Weiterhin empfiehlt es sich, eine separate Diskette sozusagen als Eingabediskette zu verwenden. Formatieren Sie also eine Diskette. Verwenden Sie aber bitte **nur DOS 2.0 oder 2.5!** Speichern Sie anschließend die DOS-Files und Turbo-Basic XL mit dem Namen »AUTORUN.SYS« auf dieser Diskette. Dann schalten Sie den Computer aus und booten die soeben angelegte Diskette. Geben Sie anschließend folgende Programmzeile ein:

```
1 BLOAD "D:PRUEFER.OBJ":POKE $BC3E,$1E:*L-:NEW
```

Speichern Sie diese Programmzeile dann mit »SAVE "D:AUTORUN.BAS"« auf Ihrer Eingabediskette. Dieses kurze Programm sorgt beim Booten des Computers dafür, daß das Prüfsummenprogramm automatisch geladen und ausgeführt wird. Achtung: **Das Prüfsummenprogramm läßt sich nicht vom DOS-Menü aus starten! Die oben aufgeführte Programmzeile muß unbedingt beim Bootvorgang ausgeführt werden.** Auf Ihrer Prüfsummer-Diskette sollten sich also folgende Files befinden:

1. DOS.SYS (entweder DOS 2.0 oder 2.5)
2. DUP.SYS (passend zum DOS.SYS-File)
3. AUTORUN.SYS (Turbo-Basic XL)
4. AUTORUN.BAS (das Initialisierungsprogramm)
5. PRUEFER.OBJ (das eigentliche Prüfsummenprogramm als Maschinencode-File)

Wenn Sie Turbo-Basic unter unserem Happy-DOS verwenden, müssen die einzelnen Files anders auf die Diskette geschrieben werden. Die folgende Liste zeigt wie:

1. Diskette formatieren mit »FO#«
2. DOS.SYS auf die Diskette schreiben mit »IN#«
3. Turbo-Basic mit »COPY AUTORUN.SYS« auf die initialisierte Diskette schreiben
4. Turbo-Basic mit »REN AUTORUN.SYS TURBO.COM« auf der Diskette umbenennen
5. Mit »JOB TURBO« das Happy-DOS veranlassen, das Turbo-Basic automatisch zu laden
6. »AUTORUN.BAS« wie in der ersten Liste beschrieben auf die Diskette schreiben
7. »PRUEFER.OBJ« auf die Diskette schreiben

Das Turbo-Basic wird dann automatisch geladen, gestartet und der Prüfsummer installiert.

Eine ähnlich Prozedur muß angewendet werden, wenn andere DOS-Versionen, wie zum Beispiel OSS CP/A 1.2, 2.1 oder DOS XL, benutzt werden. Anstelle von Punkt 5 müssen Sie allerdings von Basic aus die Befehlsfolge »OPEN #1,8,0,"D:STARTUP:EXC":? #1; "TURBO": END« eingeben. Dann wird auch in diesen DOS-Versionen Ihre Prüfsummer-Diskette automatisch gestartet.

Basic-Programme schnell und zuverlässig eingegeben

Um also unter Turbo-Basic XL den Prüfsummer verwenden zu können, brauchen Sie nur Ihre Prüfsummer-Diskette ins Laufwerk einzuschieben und Ihren Computer einzuschalten. Alles weitere erledigt der Computer von selbst. Vergessen Sie bitte nicht, sich von der neuen Prüfsummer-Version auch eine Sicherheitskopie anzulegen. Sind noch Fragen offen, wenden Sie sich bitte an die Redaktion.

(Werner Breuer/hf)

Steckbrief	
Programm:	Prüfsummer
Sprache:	Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Kassette, Diskette

```
100 REM ******************************    <LU>
110 REM *                            *    <HA>
120 REM *   PRUEFCODE fuer ATARI     *    <CU>
130 REM *                            *    <HE>
140 REM *  von  W.Kress              *    <ZZ>
150 REM *       Lindenweg 17         *    <XA>
160 REM *       7590 Achern          *    <IO>
170 REM *  (c)  by Happy-Computer    *    <DC>
180 REM *                            *    <HO>
190 REM ******************************    <MM>
200 REM                                   <AJ>
210 DIM F$(15),SED$(2),D$(504),C$(11)     <RC>
220 C$="{CTL ,}{CTL D}u7{CTL A}{CTL F})<
{CTL M}{CTL B}S"                          <IZ>
230 GRAPHICS 0:POKE 82,0:POKE 752,1       <KV>
240 POSITION 12,10:POKE 201,8             <BO>
250 ? "Bitte▴warten▴!":POKE 752,0         <DX>
260 Z=1000                                <GZ>
270 FOR I=1 TO 497 STEP 8                 <OL>
280 PSUMM=0                               <MG>
290 FOR J=0 TO 7                          <PK>
300 READ SED$                             <ZV>
310 H=ASC(SED$(1))-48                     <EQ>
320 IF H>9 THEN H=H-7                     <LT>
330 L=ASC(SED$(2))-48                     <GE>
340 IF L>9 THEN L=L-7                     <SR>
350 DEZ=H*16+L:D$(I+J)=CHR$(DEZ)          <JV>
360 PSUMM=PSUMM+DEZ                       <MV>
370 NEXT J                                <GR>
380 READ ZSUMM                            <AW>
390 IF PSUMM=ZSUMM THEN 410               <IN>
400 ? "{ESC CTL 2}DATENFEHLER▴!":LIST Z:
END                                       <YX>
410 GSUMM=GSUMM+PSUMM:Z=Z+10              <NJ>
420 NEXT I                                <FY>
430 READ ESUMM                            <SL>
440 IF ESUMM=GSUMM THEN 460               <ZO>
450 ? "{ESC CTL 2}FEHLENDE▴ODER▴DOPPELTE
▴DATA-ZEILE":END                          <IZ>
460 ? "{ESC CTL 2}"                       <EX>
470 ? "{ESC CTL <}"                       <RP>
480 POSITION 2,6:? :POKE 752,1            <UQ>
490 ? ,"{CTL Y}1{CTL Y}▴Auf▴DISKETTE▴sch
reiben{ESC CTL =}{ESC CTL =}"             <AY>
500 ? ,"{CTL Y}2{CTL Y}▴Auf▴CASSETTE▴sch
reiben{ESC CTL =}{ESC CTL =}"             <AQ>
510 ? ,"{CTL Y}3{CTL Y}▴ENDE"             <VR>
520 OPEN #1,4,0,"K":GET #1,K:CLOSE #1     <NQ>
530 K=K-48:IF K<1 OR K>3 THEN ? "{ESC CT
L 2}";:GOTO 520                           <WU>
540 POSITION 3,4+3*K:? "--->";            <FU>
550 POSITION 2,20:? "{ESC SHIFT DEL}";    <EW>
560 ON K GOTO 570,590,680                 <DI>
570 POKE 752,0:? "{ESC TAB}DATEINAME▴:▴▴
```

Listing zum Atari-Prüfsummer



```
D1:{ESC CTL +}{ESC CTL +}{ESC CTL +}{ESC
 CTL +}";                                    <KM>
580 POKE 764,44:INPUT F$:D=1:GOTO 610        <ZI>
590 F$="C:":D=0                              <MT>
600 ? ,"CASSETTE▴FERTIG▴MACHEN,▴RETURN";     <AH>
610 TRAP 660                                 <OJ>
620 OPEN #1,8,128,F$                         <YW>
630 IF D THEN ? #1;D$(1,500);                <UB>
640 IF  NOT D THEN ? #1;C$;D$(7,494)         <XK>
650 CLOSE #1:GOTO 470                        <EW>
660 ? "{ESC CTL 2}{ESC CTL -}{ESC SHIFT
DEL}{ESC TAB}I/O▴-▴FEHLER▴";PEEK(195);       <CR>
670 TRAP 40000:CLOSE #1:GOTO 480             <HQ>
680 POKE 752,0:END                           <YW>
690 REM -----------------------------        <ZP>
1000 DATA▴FF,FF,00,38,E7,39,A2,FF,1271       <EA>
1010 DATA▴BD,EB,38,9D,00,06,CA,E0,1069       <NF>
1020 DATA▴FF,D0,F5,A2,60,BD,8A,38,1349       <VW>
1030 DATA▴9D,00,01,CA,10,F7,A2,0F,800        <OL>
1040 DATA▴BD,00,E4,9D,61,01,CA,10,890        <LI>
1050 DATA▴F7,18,AD,65,01,69,01,8D,793        <MP>
1060 DATA▴6B,06,AD,66,01,69,00,8D,635        <CE>
1070 DATA▴6C,06,A9,2A,8D,65,01,A9,737        <XW>
1080 DATA▴06,8D,66,01,AD,67,01,8D,668        <VG>
1090 DATA▴30,06,AD,68,01,8D,34,06,531        <HH>
1100 DATA▴38,AD,30,02,E9,02,8D,02,657        <ZR>
1110 DATA▴06,8D,11,06,AD,31,02,E9,627        <EV>
1120 DATA▴00,8D,07,06,8D,12,06,A9,488        <DM>
1130 DATA▴45,20,7A,38,90,11,8E,26,620        <CP>
1140 DATA▴06,A9,00,20,7A,38,90,07,536        <YQ>
1150 DATA▴8E,17,06,20,01,06,18,60,330        <QZ>
1160 DATA▴A2,00,DD,1A,03,F0,08,E8,892        <CH>
1170 DATA▴E8,E8,E0,21,D0,F4,18,60,1293       <CK>
1180 DATA▴70,42,39,01,9E,06,9E,06,564        <UJ>
1190 DATA▴9E,06,75,06,9E,06,9E,06,615        <EX>
1200 DATA▴00,00,00,00,4C,2E,22,54,240        <CY>
1210 DATA▴3A,22,2C,50,45,45,4B,28,469        <TN>
1220 DATA▴31,31,33,29,2B,32,35,36,390        <LL>
1230 DATA▴2A,50,45,45,4B,28,31,31,473        <SG>
1240 DATA▴34,29,9B,01,61,45,01,04,420        <SW>
1250 DATA▴54,30,72,75,65,66,63,6F,776        <YW>
1260 DATA▴64,65,00,1A,00,00,1E,00,257        <ER>
1270 DATA▴00,1C,00,00,08,63,09,00,144        <LV>
1280 DATA▴18,15,00,62,79,00,37,0E,333        <NU>
1290 DATA▴00,2B,72,65,73,73,0F,77,622        <KV>
1300 DATA▴62,68,A9,FF,8D,30,02,A9,986        <OK>
1310 DATA▴FF,8D,31,02,A0,03,B9,00,795        <ZL>
1320 DATA▴01,99,FF,FF,88,10,F7,A0,1223       <DR>
1330 DATA▴FF,A2,05,BD,33,01,99,1A,842        <OO>
1340 DATA▴03,C8,E0,03,D0,02,A0,FF,1055       <QQ>
1350 DATA▴CA,10,F0,60,A5,77,F0,0A,1088       <OT>
1360 DATA▴A9,FF,85,26,A9,FF,85,27,1191       <LE>
1370 DATA▴C6,77,A5,73,F0,2D,A5,A1,1208       <VB>
1380 DATA▴10,04,C6,73,F0,25,A6,78,896        <NG>
1390 DATA▴BD,14,01,E6,78,C9,9B,D0,1124       <ZM>
1400 DATA▴4F,A2,00,86,78,86,73,A6,910        <WM>
1410 DATA▴A0,86,71,A6,A1,86,72,A2,1144       <QR>
1420 DATA▴9E,86,26,A2,06,86,27,E6,901        <AY>
1430 DATA▴77,D0,35,20,3E,F6,08,C9,929        <CM>
1440 DATA▴9B,D0,02,E6,73,28,60,A2,1008       <IO>
1450 DATA▴00,C9,9B,F0,26,86,CC,A6,1138       <LN>
1460 DATA▴00,86,D1,85,CD,A0,08,46,919        <OZ>
1470 DATA▴D1,90,0C,18,A2,03,B5,CD,940        <DL>
1480 DATA▴75,CA,95,CD,CA,D0,F7,06,1336       <ZZ>
1490 DATA▴CD,26,CC,88,D0,E9,E6,00,1254       <EW>
1500 DATA▴A0,01,60,A9,A4,A0,02,20,784        <HA>
1510 DATA▴CC,06,A2,03,B5,CA,95,CD,1112       <MP>
1520 DATA▴CA,D0,F9,A9,1A,20,CC,06,1096       <IN>
1530 DATA▴A5,CD,18,69,21,8D,48,01,746        <BX>
1540 DATA▴A5,D0,69,21,8D,47,01,84,856        <UQ>
1550 DATA▴D0,C8,84,00,60,85,72,84,1015       <LY>
1560 DATA▴71,86,CB,86,CC,86,CD,A0,1287       <JS>
1570 DATA▴18,18,A2,05,36,CB,CA,10,690        <RJ>
1580 DATA▴FB,38,A2,03,B5,CA,F5,6F,1211       <RY>
1590 DATA▴95,73,CA,D0,F7,90,0B,B5,1257       <GI>
1600 DATA▴74,95,CB,E8,E0,03,D0,F7,1382       <II>
1610 DATA▴E6,D0,88,D0,DC,60,E2,02,1326       <JV>
1620 DATA▴E3,02,00,38,00,00,00,00,285        <BR>
1630 DATA▴53423                              <NQ>
```

Listing zum Atari-Prüfsummer (Schluß)

Ampel-Version 1.1

Sie möchten unser neuestes Listing abtippen? Mit AMPEL wird das Eingeben von reinen Atari-Maschinenprogrammen zum Kinderspiel.

Speziell bei Maschinenprogrammen kommt es darauf an, daß jedes Byte korrekt eingegeben wird. Schon ein einziger falscher Wert führt meist dazu, daß das eingegebene Programm nicht läuft. Dies hat dann oft eine stundenlange Fehlersuche zur Folge. Deshalb haben wir für Sie »AMPEL« (Atari-Maschinen-Programm-Eingabe-Listing) entwickeln lassen. Und damit das Eintippen für Sie möglichst einfach und reibungslos verläuft, wählen wir eine spezielle Darstellungsform für Maschinenprogramme. Beachten Sie bitte, daß Sie zukünftig unbedingt AMPEL-Version 1.1 verwenden müssen, um Atari-Maschinenprogramme einzugeben. Von Basic aus lassen sich solche Listings nicht eingeben. Verwahren Sie also das Programm zu AMPEL sorgfältig und legen Sie eine zusätzliche Kopie an einen sicheren Platz.

Geben Sie zunächst das nachfolgende Listing ein. Verwenden Sie dazu den Atari-Prüfsummer. Bevor Sie AMPEL mit RUN starten, sollten Sie das eingegebene Basic-Programm auf Diskette oder Kassette speichern. Hat sich nämlich ein Fehler im Maschinensprache-Teil eingeschlichen, kann Ihr Computer abstürzen. In dem Fall müßte AMPEL erneut eingegeben werden.

Nachdem Sie das Programm mit RUN gestartet haben, werden zuerst die DATA-Werke eingePOKEt. Dieser Vorgang nimmt einige Sekunden in Anspruch. Danach müssen Sie die Länge des einzugebenden Maschinenprogramms eingeben. Diesen Wert entnehmen Sie bitte am Ende des abgedruckten Maschinenprogramms. Anschließend geben Sie den Programmnamen an, unter dem Ihr Programm gespeichert wird. Auf dem Bildschirm erscheint jetzt »0000:«, die Aufforderung, mit der Eingabe der ersten Zeile des AMPEL-Listings zu beginnen.

Betrachten wir eine AMPEL-Listing-Zeile aus der Nähe:

```
0000:00 02 00 07 06 07 20 BD <F3>
```

Die ersten vier Zahlen (hier 0000) stellen sozusagen die Zeilennummer dar. Diese Zahlen sowie den darauffolgenden Doppelpunkt brauchen Sie nicht einzugeben. Danach folgen acht zweistellige Hexadezimalzahlen, die Sie dem Listing entnehmen. Die Eingabe erfolgt ohne Betätigung der <RETURN>-Taste. Der Cursor springt selbständig von einer Position zur nächsten. Leerzeichen sowie die beiden Zeichen »<« und »>« fügt AMPEL automatisch ein. Ist die Prüfsumme korrekt – sie muß auch von Ihnen eingegeben werden – springt der Cursor in die nächste Zeile. Falls nicht, ertönt ein akustisches Signal und alle acht Werte, inklusive der Prüfsumme, müssen nochmals eingegeben werden. Dadurch wir gewährleistet, daß wirklich alle eingegebenen Hexadezimalzahlen im Listing stimmen und sich keine unbemerkten Fehler einschleichen können.

Außer den Hex-Tasten (<0> bis <9> und <A> bis <F>), wird nur noch <DELETE BACKSPACE> zum Korrigieren des letzten Zeichens und <CONTROL S> abgefragt.

Zwischenspeichern

Da Sie ein umfangreiches Maschinenprogramm sicher nicht an einem Tag eintippen möchten, können Sie zu jedem Zeitpunkt mit <CONTROL S> zwischenspeichern. Dazu betätigen Sie einfach <CONTROL S> und der Code wird unter dem zuvor eingegebenen Programmnamen gespeichert. Weiterhin wird automatisch die jeweils zuletzt gespeicherte Version in »BACKUP.OBJ« umbenannt. Zuvor wird natürlich eine ältere »BACKUP.OBJ«-Datei gelöscht. Danach können Sie den Computer ausschalten oder AMPEL mit <RESET> verlassen.

Wenn Sie das nächstemal AMPEL starten, muß wieder die Programmlänge und der entsprechende Name, mit dem Sie Ihre letzte Version gespeichert haben, eingegeben werden. Existiert bereits eine Datei mit dem entsprechenden Namen, wird die Datei automatisch geladen. Weiterhin wird noch die Zeilennummer auf dem Bildschirm angezeigt, ab der Sie mit der Eingabe der Hex-Werte fortfahren müssen.

Speichern Sie lieber öfter mit <CONTROL+S>, um auch gegen eventuelle Stromausfälle gewappnet zu sein. Weiterhin empfiehlt es sich aus Sicherheitsgründen, zwei Disketten für Daten vorzusehen. Wenn Sie das komplette Programm eingegeben haben, speichert der Computer nach Eingabe des letzten Bytes das Programm auf Diskette und meldet sich danach mit der READY-Meldung.

Wenn Sie Ihr Maschinenprogramm anschließend laden möchten, erfolgt dies vom DOS-Menü aus. Wählen Sie hier die L-Funktion und geben Sie anschließend den entsprechenden Programmnamen ein.

Besondere Hinweise für Kassettenrecorder-Besitzer

Damit AMPEL auch einwandfrei mit einem Kassettenrecorder funktioniert, müssen folgende Programmzeilen geändert werden:

```
290 F$="C:"   :AD=0
300 ? " Altes File laden J/N "  ;
310 INPUT FR$
320 IF FR$<>"J" THEN 510
420 TRAP 510:OPEN #1,4,128,F$
590 TRAP 660:CLOSE#1:OPEN #1,8,128,F$
```

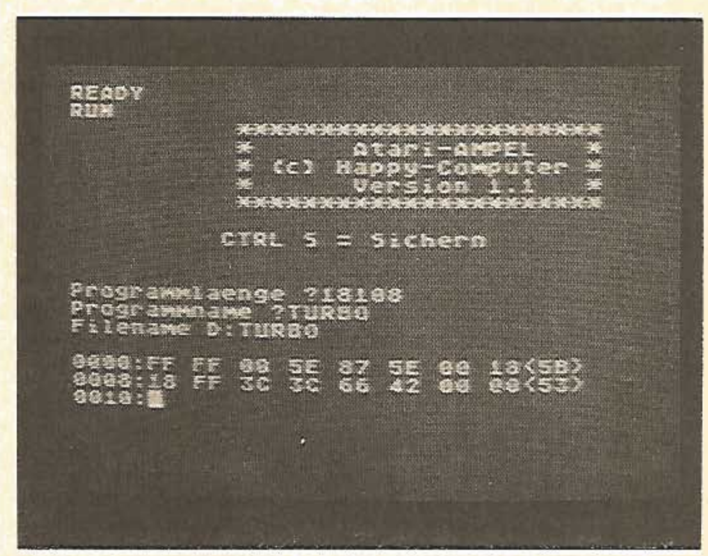
Die Zeilen 330 bis 360 entfallen, ebenso die Zeilen 450, 600 und 610.

Speichern Sie anschließend die geänderte AMPEL-Version mit CSAVE. Somit haben Sie das benötigte Eingabeprogramm bereits vorliegen.

Sicherheit über alles

Mit <CONTROL S> kann zu jedem Zeitpunkt gespeichert werden. Daraufhin erklingt ein doppelter Signalton. Betätigen Sie dann noch die <RECORD>- und <PLAY>-Tasten des Recorders und anschließend die <RETURN>-Taste. Der Speichervorgang beginnt. Wenn Sie dann mit einem anderen Programm arbeiten möchten, gelangen Sie mit <RESET> zurück ins Basic.

Wenn Sie zu einem späteren Zeitpunkt mit der Arbeit fortfahren möchten, laden Sie wieder zuerst AMPEL, starten es mit RUN, geben die Programmlänge ein und beantworten die Frage »Altes File laden J/N« mit J. Daraufhin meldet sich der Computer wieder mit einem Brummton. Spulen Sie das Band an den Anfang zurück, betätigen Sie dann noch die <PLAY>-Taste und <RETURN>. Jetzt wird der zuletzt eingegebene Programmteil geladen. Anschließend meldet sich der Atari-Computer wieder mit der Zeilennummer, ab der Sie mit der Eingabe fortfahren müssen.



So sieht der Bildschirm bei der Eingabe von Maschinenprogrammen aus. Zwischenspeichern kann man jederzeit mit der Tastenkombination <CONTROL+S>.

Um später das eingegebene Programm zu starten, muß stets zuerst das »Kassetten-Hilfsprogramm« geladen werden. Nach dem Ladevorgang meldet sich der Computer mit dem Brummton. Anschließend legen Sie bitte Ihre Kassette mit Ihrem Maschinenprogramm in den Recorder ein, drücken die <PLAY>-Taste des Recorders und während Sie die <START>- und <OPTION>-Tasten des Computers drücken, schalten Sie die Stromversorgung ein. Nach dem Brummton nur noch <RETURN> drücken und Ihr Programm wird geladen und anschließend automatisch gestartet.

Programme schnell geladen

Speziell bei Kassettenrecordern kann es Probleme beim Laden von Maschinenprogrammen geben. Deshalb sollten Sie stets mehrere Sicherheitskopien von Ihren Programmen machen. Hinzu kommt, daß das Laden von Kassette sehr viel Zeit beanspruchen kann, da die Übertragungsgeschwindigkeit sehr langsam ist. Besitzer einer Diskettenstation haben es hier viel besser. Sollten Sie mit AMPEL auf Schwierigkeiten stoßen, wenden Sie sich an die Redaktion.

(Frank Ostrowski/Werner Breuer/hf)

Steckbrief	
Programm:	AMPEL
Sprache:	Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Kassette, Diskette



```
100 ? ,"▴***********************"              <HI>
110 ? ,"▴*▴▴▴▴▴▴▴Atari-AMPEL▴▴▴*"              <XO>
115 ? ,"▴*▴(c)▴Happy-Computer▴*"               <JT>
117 ? ,"▴*▴▴▴▴▴▴▴Version▴1.1▴▴▴*"              <OE>
120 ? ,"▴***********************"              <HM>
130 ?                                          <JU>
140 ? ,"CTRL▴S▴=▴Sichern"                      <ZY>
150 ?                                          <JY>
160 ? "Einen▴Moment▴Geduld▴bitte.";            <CL>
190 POKE 16,64:POKE 53774,64                   <OQ>
230 DIM F$(15),FB$(15),FR$(27)                 <WK>
240 DIM ML$(261),CIO$(83)                      <MH>
250 GOSUB 1000                                 <QT>
260 ? CHR$(156)                                <DG>
270 ? "Programmlaenge▴";:INPUT L               <EH>
280 DIM A$(L+8)                                <GW>
290 ? "Programmname▴";:INPUT F$                <IR>
300 IF F$="" THEN 290                          <MA>
310 F=0:FOR I=1 TO LEN(F$):F=F+(F$(I,I)=
":"):NEXT I                                    <IR>
320 IF F=0 THEN FR$="D:":FR$(3)=F$:F$=FR
$                                              <UE>
330 ? "Filename▴";F$                           <TA>
340 FB$="D:BACKUP.OBJ"                         <NW>
350 FR$=F$:FR$(LEN(FR$)+1)=","                 <XT>
360 FR$(LEN(FR$)+1)=FB$(3)                     <IO>
420 AD=0:TRAP 510:OPEN #1,4,0,F$               <QM>
430 E=USR(ADR(CIO$),1,7,ADR(A$),L)             <ZH>
440 IF E=1 THEN ? "Schon▴fertig":END           <IN>
450 IF E<>136 THEN ? "Disk-Read-Error▴";
E:END                                          <QL>
460 AD=PEEK(856)+256*PEEK(857)                 <QT>
470 AD=8*INT(AD/8):TRAP 65535                  <VQ>
510 CD=USR(ADR(ML$),ADR(A$),AD)                <XM>
520 ON CD GOSUB 590,690,710                    <PW>
530 AD=AD+8*(CD=0)                             <SV>
540 IF AD<=L THEN 510                          <WK>
550 AD=L:GOSUB 590                             <NI>
560 POKE 16,192:POKE 53774,192:END             <XR>
590 TRAP 600:CLOSE #1:XIO 33,#1,0,0,FB$        <MD>
600 TRAP 610:CLOSE #1:XIO 32,#1,0,0,FR$        <IF>
610 TRAP 660:CLOSE #1:OPEN #1,8,0,F$           <MU>
620 E=USR(ADR(CIO$),1,11,ADR(A$),AD)           <LC>
630 CLOSE #1                                   <LI>
640 IF E<>1 THEN ? "Disk-Error▴";E             <ZN>
650 TRAP 65535:RETURN                          <FB>
660 CLOSE #1                                   <LO>
670 ? "Disk-Error▴";PEEK(195):RETURN           <XX>
690 ? CHR$(253):? "Pruefsummenfehler"          <PO>
700 RETURN                                     <MR>
710 ? "Fehler!!!!";:LIST 510:END               <CW>
1000 Q=0                                       <WO>
1010 FOR I=1536 TO 1592                        <NU>
1020 READ A:Q=Q+A:POKE I,A                     <KX>
1030 NEXT I                                    <EY>
1040 IF Q<>4196 THEN ? "Data▴Error":LIST
 1300,1390:END                                 <JV>
1050 Q=0                                       <XD>
1060 FOR I=1 TO 261                            <RZ>
1070 READ A                                    <QI>
1080 IF A>255 THEN GOSUB 1190                  <EL>
1090 ML$(I)=CHR$(A):Q=Q+A                      <MU>
1100 NEXT I                                    <ER>
1110 READ A:IF Q<>A THEN 1200                  <FZ>
1120 Q=0                                       <WW>
1130 FOR I=1 TO 83                             <JT>
1140 READ A:Q=Q+A:CIO$(I)=CHR$(A)              <QL>
1150 NEXT I                                    <FG>
1160 IF Q<>10894 THEN ? "Data▴Error▴";:L
IST 1950,2080:END                              <MP>
1170 RETURN                                    <PQ>
1190 IF Q=A THEN READ A:Q=0:RETURN             <VA>
1200 ? "Data▴Error"                            <GO>
1210 IF A=7931 THEN LIST 1430,1530             <IU>
1220 IF A=6585 THEN LIST 1540,1640             <PX>
1230 IF A=8135 THEN LIST 1650,1750             <RD>
1240 IF A=7184 THEN LIST 1760,1860             <XK>
1250 IF A=2222 THEN LIST 1870,1910             <OU>
1300 DATA▴173,37,228,72,173,36                 <YX>
1310 DATA▴228,72,96,72,74,74                   <WJ>
1320 DATA▴74,74,32,20,6,104                    <ZA>
1330 DATA▴41,15,9,48,201,58                    <JO>
1340 DATA▴144,2,105,6,168,173                  <OH>
1350 DATA▴71,3,72,173,70,3                     <DW>
1360 DATA▴72,152,96,50,31,30                   <CG>
1370 DATA▴26,24,29,27,51,53                    <IL>
1380 DATA▴48,63,21,18,58,42                    <KV>
1390 DATA▴56,190,52                            <PY>
1430 DATA▴169,0,133,213,104,201                <HX>
1440 DATA▴2,240,16,170,240,5                   <YR>
1450 DATA▴104,104,202,208,251,169              <RJ>
1460 DATA▴3,44,169,1,133,212                   <BU>
1470 DATA▴96,104,133,215,104,133               <YK>
1480 DATA▴214,104,133,217,104,133              <GO>
1490 DATA▴216,24,101,214,133,214               <UF>
1500 DATA▴165,217,101,215,133,215              <IU>
1510 DATA▴169,155,32,28,6,165                  <SR>
1520 DATA▴217,32,9,6,165,216                   <LG>
1530 DATA▴7931                                 <MS>
1540 DATA▴32,9,6,169,58,32                     <OX>
1550 DATA▴28,6,169,0,133,218                   <JQ>
1560 DATA▴173,252,2,41,191,162                 <NS>
1570 DATA▴17,221,39,6,240,5                    <DP>
1580 DATA▴202,16,248,48,239,134                <GQ>
1590 DATA▴219,141,252,2,32,0                   <UW>
1600 DATA▴6,41,127,32,28,6                     <GB>
1610 DATA▴165,219,201,16,240,168               <IB>
1620 DATA▴144,40,165,218,240,205               <CF>
1630 DATA▴74,176,5,169,126,32                  <RS>
1640 DATA▴6585                                 <PH>
1650 DATA▴28,6,160,4,70,220                    <YY>
1660 DATA▴102,221,102,222,102,223              <VE>
1670 DATA▴102,224,102,225,102,226              <DY>
1680 DATA▴102,227,102,228,136,208              <QR>
1690 DATA▴235,198,218,184,80,178               <BA>
1700 DATA▴10,10,10,10,160,4                    <LO>
1710 DATA▴10,38,228,38,227,38                  <UU>
1720 DATA▴226,38,225,38,224,38                 <YP>
1730 DATA▴223,38,222,38,221,38                 <RU>
1740 DATA▴220,136,208,234,230,218              <OG>
1750 DATA▴8135                                 <LZ>
1760 DATA▴165,218,201,18,240,19                <WO>
1770 DATA▴201,16,208,4,169,60                  <MK>
1780 DATA▴208,5,74,176,206,169                 <EG>
1790 DATA▴32,32,28,6,184,80                    <JZ>
1800 DATA▴198,165,217,10,101,216               <ZP>
1810 DATA▴10,101,220,10,101,221                <UO>
1820 DATA▴10,101,222,10,101,223                <YP>
1830 DATA▴10,101,224,10,101,225                <CQ>
1840 DATA▴10,101,226,10,101,227                <GR>
1850 DATA▴197,228,208,18,169,62                <NJ>
1860 DATA▴7184                                 <NS>
1870 DATA▴32,28,6,160,7,185                    <LL>
1880 DATA▴220,0,145,214,136,16                 <KJ>
1890 DATA▴248,169,0,44,169,2                   <NU>
1900 DATA▴133,212,96                           <CA>
1910 DATA▴2222                                 <HM>
1950 DATA▴104,201,4,208,66,104                 <KE>
1960 DATA▴104,170,104,104,133,212              <AX>
1970 DATA▴104,133,214,104,133,213              <DH>
1980 DATA▴104,133,216,104,133,215              <HO>
1990 DATA▴160,134,138,201,8,176                <AX>
2000 DATA▴33,10,10,10,10,170                   <LS>
2010 DATA▴165,212,157,66,3,165                 <VR>
2020 DATA▴213,157,68,3,165,214                 <TO>
2030 DATA▴157,69,3,165,215,157                 <AD>
2040 DATA▴72,3,165,216,157,73                  <SE>
2050 DATA▴3,32,86,228,132,212                  <HM>
2060 DATA▴169,0,133,213,96,160                 <QU>
2070 DATA▴10,170,240,244,104,104               <SB>
2080 DATA▴202,208,251,240,237                  <JQ>
```

Eingabehilfe »AMPEL« für Programme in Maschinensprache. Bitte mit dem Prüfsummer von Seite 60 eingeben.

Auf die Kassette, fertig, los!

Benutzer eines Kassettenrecorders dürfen endlich aufatmen. Auch sie können nun die Vorteile des »Turbo-Basic« voll nutzen.

Alle (Atari-)Welt spricht vom Turbo-Basic und erfreut sich der immensen Vorteile und erweiterten Fähigkeiten gegenüber dem »Atari-Basic« – vorausgesetzt, man verfügt über ein Diskettenlaufwerk. Wer indes nur einen Kassettenrecorder sein eigen nennt, stand bislang im Regen. Die Ladeprozedur des Turbo-Basic war nämlich ausschließlich auf Diskettenbetrieb ausgelegt. Diesem Mißstand macht das Programm »Turbo-Load« jetzt ein Ende. Sollten Sie der Vorzüge des Turbo-Basic noch nicht gewahr sein, hier ein paar Appetitanreger:

Der Turbo-Basic-Interpreter
- ist voll kompatibel zum Atari-Basic
- macht auch Atari-Basic-Programme wesentlich schneller
- erweitert den Befehlssatz um so wichtige Elemente wie neue Schleifenformen (DO-UNTIL, WHILE-WEND etc.) und Grafik-Kommandos (beispielsweise zum Zeichnen von Kreisen und Rechtecken sowie zum Füllen beliebiger Flächen).

Eine vollständige Aufstellung des Befehlsvorrats und der Handhabung des Interpreters entnehmen Sie bitte dem Beitrag »Basic, schnell wie der Wind« ab Seite 68 dieser Ausgabe.

Geben Sie zunächst das Turbo-Load-Listing mit AMPEL ein und speichern Sie es. Auf der gleichen Kassette, genau hinter Turbo-Load, speichern Sie dann das ebenfalls eingegebene Turbo-Basic. Zum Schutz vor unwiederbringlichen Datenverlusten fertigen Sie mindestens eine Arbeitskopie von dieser Turbo-Basic-Kassette. Zum Einsatz des Turbo-Basic spulen Sie nun die Kassette auf den Anfang von Turbo-Load und schalten den Computer ein, während Sie gleichzeitig die Tasten <OPTION> und <START> gedrückt halten. Hören Sie den Summton, lassen Sie die Tasten wieder los und starten den Recorder. Nach einer kurzen Ladezeit fordert Turbo-Load Sie auf, die Kassette an den Beginn der Turbo-Basic-Datei zu spulen, die Taste <PLAY> des Recorders und abschließend <RETURN> auf der Computer-Tastatur zu drücken. Eine knappe Viertelstunde später meldet sich Turbo-Basic zur Stelle.

Der Interpreter ist fest im Computer installiert, auch die <RESET>-Taste bringt ihn nicht aus der Ruhe. Geben Sie den Befehl »? FRE(0)« ein. Wenn der Computer die Zahl »34018« ausgibt, ist alles in Ordnung. (ja)

Steckbrief	
Programm:	Turbo-Load
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Kassette

```
Programmname :D:TURBOLOA.BIN
Laenge       :904 Bytes

0000:00 08 00 07 21 07 A9 3C <24>
0008:8D 02 D3 A9 21 85 0C A9 <14>
0010:07 85 0D A9 22 85 0A A9 <01>
0018:07 85 0B 18 60 00 00 00 <DA>
0020:00 60 A2 00 BD 00 E0 9D <AF>
0028:00 0C BD 00 E1 9D 00 0D <49>
0030:BD 00 E3 9D 00 0F E8 D0 <90>
0038:EB A2 00 BD 36 09 9D 00 <F0>
0040:0E E8 E0 88 D0 F5 A9 0C <87>
0048:8D C5 02 A9 10 8D C6 02 <FA>
0050:8D F0 02 A9 0C 8D F4 02 <EB>
0058:A2 00 A9 0B 8D 42 03 A9 <27>
0060:BE 8D 44 03 A9 09 8D 45 <E4>
0068:03 A9 A7 8D 48 03 A9 00 <51>
0070:8D 49 03 20 56 E4 A2 10 <C0>
0078:A9 03 9D 42 03 A9 85 9D <A8>
0080:44 03 A9 0A 9D 45 03 A9 <EA>
0088:04 9D 4A 03 A9 80 9D 4B <38>
0090:03 20 56 E4 A2 10 A9 07 <AA>
0098:9D 42 03 A9 1D 9D 44 03 <C9>
00A0:A9 07 9D 45 03 A9 02 9D <7A>
00A8:48 03 A9 00 9D 49 03 20 <D6>
00B0:56 E4 10 0A C0 88 F0 03 <7A>
00B8:4C 3E 08 4C 68 08 A9 FF <D7>
00C0:CD 1D 07 D0 05 CD 1E 07 <EE>
00C8:F0 CA AD 1E 07 C9 C0 90 <57>
00D0:09 38 AD 1E 07 E9 50 8D <04>
00D8:1E 07 A2 10 A9 1F 9D 44 <3F>
00E0:03 A9 07 9D 45 03 20 56 <FB>
00E8:E4 10 03 4C 3E 08 AD 20 <81>
00F0:07 C9 C0 90 09 38 AD 20 <56>
00F8:07 E9 50 8D 20 07 AD 1F <10>
0100:07 38 ED 1D 07 8D 1F 07 <B2>
0108:AD 20 07 ED 1E 07 8D 20 <DE>
0110:07 EE 1F 07 D0 03 EE 20 <EB>
0118:07 A2 10 AD 1D 07 9D 44 <55>
0120:03 AD 1E 07 9D 45 03 AD <B4>
0128:1F 07 9D 48 03 AD 20 07 <95>
0130:9D 49 03 20 56 E4 30 03 <DA>
0138:4C 94 07 4C 3E 08 20 5D <9A>
0140:08 A2 00 A9 65 8D 44 03 <66>
0148:A9 0A 8D 45 03 A9 20 8D <AD>
0150:48 03 A9 00 8D 49 03 20 <FB>
0158:56 E4 4C 22 07 A2 10 A9 <51>
0160:0C 9D 42 03 20 56 E4 60 <8A>
0168:AE 24 E4 AC 25 E4 E8 D0 <DA>
0170:01 C8 8E F8 24 8C F9 24 <9B>
0178:A9 00 8D 0E D4 78 A9 FE <15>
0180:8D 01 D3 A9 AB 8D FA FF <FA>
0188:A9 24 8D FB FF A9 D0 8D <91>
0190:FE FF A9 24 8D FF FF A9 <B8>
0198:CC 20 AD 08 A9 E0 20 AD <74>
01A0:08 A9 40 8D 0E D4 58 20 <30>
01A8:D3 08 4C 47 60 85 D5 A0 <62>
01B0:00 84 D4 A2 04 A9 FF 8D <79>
01B8:01 D3 B1 D4 99 00 21 C8 <12>
01C0:D0 F8 CE 01 D3 B9 00 21 <10>
01C8:91 D4 C8 D0 F8 E6 D5 CA <D9>
01D0:D0 E3 60 A9 FE 8D 01 D3 <01>
01D8:A9 00 85 D6 A9 70 85 D7 <22>
01E0:A9 F7 85 D8 A9 7B 85 D9 <1A>
01E8:20 12 09 A9 00 85 D6 A9 <81>
01F0:88 85 D7 A9 F9 85 D8 A9 <83>
01F8:8F 85 D9 20 12 09 A9 00 <D6>
0200:85 D6 A9 94 85 D7 A9 F1 <A6>
0208:85 D8 A9 AF 85 D9 20 12 <BE>
0210:09 60 A5 D7 18 69 50 85 <3A>
0218:DB A0 00 84 DA B1 D6 91 <CC>
0220:DA C4 D8 D0 06 A5 D7 C5 <4F>
0228:D9 F0 0A C8 D0 EF E6 D7 <2F>
0230:E6 DB 4C 1D 09 60 00 18 <FA>
0238:18 FF 3C 3C 66 42 00 00 <7F>
0240:00 06 66 7E 76 66 00 00 <C7>
0248:18 3C 66 7E 66 66 00 3E <22>
0250:73 33 3E 30 30 78 00 7C <E0>
0258:37 33 3E 30 30 78 00 00 <AA>
0260:60 63 36 3C 39 63 00 01 <C7>
0268:01 0F 03 03 F6 E4 00 80 <00>
0270:80 F0 C0 C0 60 20 00 01 <A8>
0278:03 06 06 06 03 01 00 E1 <7D>
0280:33 06 06 06 33 E1 00 C7 <84>
0288:63 33 36 36 66 CE 00 0E <69>
0290:9C 9C F6 66 66 67 00 7E <35>
0298:33 33 3E 30 30 78 00 67 <53>
02A0:66 66 66 66 66 3C 00 FF <B3>
02A8:31 31 31 31 31 7B 00 FB <3D>
02B0:81 81 F1 81 81 FB 00 F8 <BB>
02B8:8C 8C F8 B0 98 CC 7D 9B <33>
02C0:20 20 20 20 20 20 20 20 <A0>
02C8:20 20 20 20 20 20 00 01 <69>
02D0:02 03 04 05 06 07 9B 20 <08>
02D8:20 20 20 20 20 20 20 20 <D8>
02E0:20 20 20 20 08 09 0A 0B <66>
02E8:0C 0D 0E 0F 10 9B 9B 20 <2D>
02F0:20 20 20 20 20 20 20 20 <70>
02F8:20 20 54 55 52 42 4F 20 <BC>
0300:4C 4F 41 44 20 31 2E 30 <9E>
0308:9B 20 20 20 20 20 20 20 <CB>
0310:28 63 29 20 31 39 38 37 <FB>
0318:20 20 48 65 6E 72 69 6B <F8>
0320:20 46 69 73 63 68 9B 9B <76>
0328:9B 20 20 20 20 20 20 20 <AB>
0330:20 20 54 55 52 42 4F 20 <7E>
0338:42 41 53 49 43 20 58 4C <3D>
0340:20 31 2E 35 9B 20 20 20 <71>
0348:20 20 20 28 63 29 20 31 <15>
0350:39 38 35 20 46 72 61 6E <90>
0358:6B 20 4F 73 74 72 6F 77 <57>
0360:73 6B 69 9B 9B 9B 46 65 <6A>
0368:68 6C 65 72 20 62 65 69 <9D>
0370:6D 20 4C 61 64 65 6E 20 <F5>
0378:76 6F 6E 20 4B 61 73 73 <BA>
0380:65 74 74 65 21 43 00 00 <5F>
```

Listing. »Turbo-Load«

Wenn die Kassette mit der Diskette

Mit dem hier vorgestellten Programm lassen sich alle für Diskette gedachten Maschinenprogramme von Kassette laden und starten.

Für uns ist das schon ein Problem, die Sache mit den Programmen auf Diskette. Fast ausnahmslos alle Programme, die wir im Heft abdrucken, bekommen wir auf Diskette zugesandt. Das allein wäre nicht weiter schlimm. Nur, meistens sind die Programme so ausgelegt, daß sie nur mit der Diskettenstation zusammenarbeiten. Ganz schlimm wird es, wenn wir Maschinenprogramme erhalten. Es ist fast unmöglich, die Programme so umzuschreiben, daß sie sowohl mit Kassette als auch mit Diskette funktionieren. Außerdem müßten wir dann immer zwei Versionen abdrucken, einmal für Kassetten- und einmal für Diskettenbetrieb. Da wir den Platz im Heft lieber mit anderen Programmen oder Beiträgen füllen wollen, haben wir uns eine elegante Lösung einfallen lassen, die Nachteile weder für den Kassetten- noch für den Diskettenbetrieb mit sich bringt.

Tippen Sie das Programm mit AMPEL ab und speichern Sie es auf Ihrer Kassette. Unser »COS«, »Cassetten Operations System«, wird geladen, indem Sie den Computer ausschalten, die <START>-Taste drücken, je nachdem, ob Sie Basic benötigen oder nicht, die <OPTION>-Taste betätigen und den Computer dann einschalten. Er meldet sich mit einem Hupton. Spulen Sie jetzt die Kassette auf den Anfang von COS, schalten den Recorder auf »PLAY« und drücken eine Taste am Computer. COS wird nun geladen. Das Programm beansprucht acht Blöcke auf der Kassette und ist somit schnell im Speicher. COS meldet sich mit einem Menü.

Drei Funktionen hat COS. Die wichtigste ist die Funktion <L>, mit der ein File von der Kassette geladen und gestartet wird. Drücken Sie diese Taste, fordert der Computer Sie mit einem Hupton auf, die Kassette auf den Anfang des zu ladenden Files zu spulen, ihn wieder auf »PLAY« zu schalten und eine Taste zu betätigen. »Turbo-Basic« läßt sich mit COS leider nicht laden, da Turbo-Basic eine spezielle Laderoutine für den Diskettenbetrieb eingebaut hat, die mit Kassette nicht funktioniert.

Mit der zweiten Funktion <C> wie »Cartridge« wird zu einem eingesteckten Modul gesprungen beziehungsweise das Basic gestartet. Verwenden Sie diese Funktion nicht, wenn Sie Basic nicht eingeschaltet haben. Der Computer hängt sich sonst auf, und Sie müssen ihn mit <RESET> wieder ins Diesseits der Bits und Bytes zurückrufen. Diese Funktion wird auch zusammen mit dem Prüfsummer verwendet. Nachdem Sie den Prüfsummer von Kassette geladen haben, müssen Sie irgendwie ins Basic kommen, um den Prüfsummer benutzen zu können. Dies geschieht mit der Funktion <C>.

Als letztes steht die Funktion <S> zur Verfügung, mit der ein geladenes Programm nochmals gestartet wird. Nützlich ist dies bei verschiedenen kleinen Demonstrationsprogrammen, die nach getaner Arbeit wieder ins COS zurückkehren. (hf)

Steckbrief	
Programm:	COS
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Kassette

```
Programmname :COS.BIN
Länge        :808 Bytes
-----------------------------------

0000:00 07 00 07 3D 07 A9 3C <C7>
0008:8D 02 D3 A9 3D 85 0C A9 <F1>
0010:07 85 0D A9 48 85 0A A9 <32>
0018:07 85 08 A9 00 8D 44 02 <A9>
0020:A9 47 8D 38 07 8D 36 07 <99>
0028:A9 07 8D 39 07 8D 37 07 <C7>
0030:18 60 00 00 00 00 00 00 <14>
0038:00 00 00 00 00 A9 21 8D <AE>
0040:E7 02 A9 0A 8D E8 02 60 <59>
0048:A9 01 85 09 A2 00 A9 0B <67>
0050:8D 42 03 8D F0 02 A9 48 <79>
0058:8D 44 03 A9 09 8D 45 03 <B2>
0060:A9 99 8D 48 03 A9 00 8D <2C>
0068:49 03 20 56 E4 20 0A 09 <73>
0070:AD 3A 07 C9 4C F0 1F C9 <EC>
0078:53 D0 03 4C 96 08 C9 43 <64>
0080:D0 EB A2 00 8E F0 02 8A <B2>
0088:8D 48 03 8D 49 03 A9 7D <BF>
0090:20 56 E4 6C FE BF A9 47 <FE>
0098:8D 38 07 8D 36 07 A9 07 <35>
00A0:8D 39 07 8D 37 07 A9 00 <8E>
00A8:8D 3B 07 8D 3C 07 A2 10 <31>
00B0:A9 03 9D 42 03 A9 1F 9D <33>
00B8:44 03 A9 0A 9D 45 03 A9 <02>
00C0:04 9D 4A 03 A9 80 9D 4B <6F>
00C8:03 20 56 E4 A9 47 8D E0 <B6>
00D0:02 A9 07 8D E1 02 A2 10 <48>
00D8:A9 07 9D 42 03 A9 32 9D <62>
00E0:44 03 A9 07 9D 45 03 A9 <99>
00E8:02 9D 48 03 A9 00 9D 49 <19>
00F0:03 20 56 E4 10 0A C0 88 <11>
00F8:D0 03 4C 8A 08 4C C1 08 <BE>
0100:A9 FF CD 32 07 D0 0B CD <AD>
0108:33 07 D0 06 8D 3C 07 4C <8F>
0110:D6 07 AD 3C 07 D0 03 4C <62>
0118:E8 08 AD 3B 07 D0 0C AD <95>
0120:E2 02 8D 36 07 AD E3 02 <3F>
0128:8D 37 07 A2 10 A9 07 8D <77>
0130:3B 07 9D 42 03 A9 34 9D <09>
0138:44 03 A9 07 9D 45 03 A9 <4F>
0140:02 9D 48 03 A9 00 9D 49 <AB>
0148:03 20 56 E4 30 73 EE 34 <38>
0150:07 D0 03 EE 35 07 A2 10 <22>
0158:A9 07 9D 42 03 AD 32 07 <62>
0160:9D 44 03 AD 33 07 9D 45 <F2>
0168:03 AD 34 07 38 ED 32 07 <EE>
0170:9D 48 03 AD 35 07 ED 33 <E0>
0178:07 9D 49 03 20 56 E4 30 <CC>
0180:40 20 87 08 4C CC 07 6C <27>
0188:E0 02 AD E2 02 8D 38 07 <F8>
0190:AD E3 02 8D 39 07 20 B6 <17>
0198:08 20 B0 08 A9 47 CD E2 <1B>
01A0:02 D0 0A A9 07 CD E3 02 <96>
01A8:D0 03 20 B3 08 4C 48 07 <89>
01B0:6C 38 07 6C 36 07 A2 10 <B9>
01B8:A9 0C 9D 42 03 20 56 E4 <F3>
01C0:60 20 B6 08 A2 00 A9 0B <A3>
01C8:8D 42 03 A9 E1 8D 44 03 <5F>
01D0:A9 09 8D 45 03 A9 21 8D <F9>
01D8:48 03 A9 00 8D 49 03 20 <84>
01E0:56 E4 20 0A 09 4C 48 07 <38>
01E8:20 B6 08 A2 00 A9 02 8D <B7>
01F0:44 03 A9 0A 8D 45 03 A9 <37>
01F8:1D 8D 48 03 A9 00 8D 49 <4D>
0200:03 20 56 E4 20 0A 09 4C <1D>
0208:48 07 A2 20 A9 03 9D 42 <FC>
0210:03 A9 20 9D 44 03 A9 0A <47>
0218:9D 45 03 A9 04 9D 4A 03 <4F>
0220:20 56 E4 A9 07 9D 42 03 <C6>
0228:A9 3A 9D 44 03 A9 07 9D <4C>
0230:45 03 A9 01 9D 48 03 A9 <71>
0238:00 9D 49 03 20 56 E4 A9 <87>
0240:0C 9D 42 03 20 56 E4 60 <A8>
0248:7D 48 61 70 70 79 2D 43 <AA>
0250:6F 6D 70 75 74 65 72 20 <84>
0258:43 4F 53 20 20 20 41 54 <54>
0260:41 52 49 20 56 65 72 73 <C0>
0268:69 6F 6E 20 31 2E 30 43 <11>
0270:6F 70 79 72 69 67 68 74 <03>
0278:20 28 63 29 20 31 39 38 <65>
0280:37 20 62 79 20 48 65 6E <66>
0288:72 69 6B 20 46 69 73 63 <A7>
0290:68 9B 9B 5B 4C 5D 20 46 <12>
0298:69 6C 65 20 76 6F 6E 20 <85>
02A0:43 61 73 73 65 74 74 65 <5A>
02A8:20 6C 61 64 65 6E 9B 5B <5F>
02B0:53 5D 20 47 65 6C 61 64 <29>
02B8:65 6E 65 6E 20 46 69 6C <B3>
02C0:65 20 73 74 61 72 74 65 <1D>
02C8:6E 9B 5B 43 5D 20 47 65 <67>
02D0:68 65 20 7A 75 72 20 43 <7E>
02D8:61 72 74 72 69 64 67 65 <63>
02E0:9B 9B 9B 46 65 68 6C 65 <85>
02E8:72 20 62 65 69 6D 20 4C <C9>
02F0:61 64 65 6E 20 76 6F 6E <C5>
02F8:20 4B 61 73 73 65 74 74 <1E>
0300:65 9B 9B 9B 46 65 68 6C <6D>
0308:65 72 20 2D 20 6B 65 69 <F6>
0310:6E 20 44 4F 53 20 32 2E <6E>
0318:30 20 46 69 6C 65 9B 43 <E8>
0320:4B 00 00 00 00 00 00 00 <92>
```

Listing. »COS«, mit AMPEL eingeben

Basic, schnell wie der Wind – mit dem Turbo-Basic-XL-Interpreter

Ist Ihnen Atari-Basic zu langsam? Oder wünschen Sie sich zusätzliche Befehle? Dann ist unser Turbo-Basic-Interpreter genau das richtige Programm für Sie.

Zugegeben, unser Interpreter ist ein wenig umfangreich. Immerhin müssen insgesamt 18108 Byte erst einmal eingetippt sein. Eine Aufgabe, die sicher einige Stunden in Anspruch nimmt. Haben Sie aber dann den Interpreter auf Diskette vorliegen, sparen Sie sich zukünftig viel Zeit für die Entwicklung neuer Programme. Erstens stehen Ihnen sämtliche Befehle des normalen Atari-Basic zur Verfügung, und zweitens gewinnen Sie durch die neuen Befehle viel Zeit und vor allem Speicherplatz.

Aber Turbo-Basic XL spart nicht nur Zeit bei der Programmierung, sondern auch bei der Ausführung von Programmen. Es macht seinem Namen als schnellstes derzeit erhältliches Basic für Atari-Computer nämlich alle Ehre. Wir haben unseren Interpreter dem normalen Basic und verschiedenen Basic-Compilern gegenübergestellt. Turbo-Basic XL siegte in jeder Hinsicht.

Damit Sie mit Turbo-Basic XL und dem dazugehörigen Compiler arbeiten können, benötigen Sie unbedingt einen Atari 800XL oder 130XE. Besitzer der älteren Computer, also Atari 400 und 800, können den Interpreter leider nicht benutzen. In der abgedruckten Version setzt Turbo-Basic XL mindestens 64 KByte RAM voraus.

Beachten Sie bitte: Turbo-Basic XL muß unbedingt mit AMPEL (Atari-Maschinen-Programm-Eingabe-Listing) eingegeben werden. Von Basic aus läßt sich Turbo-Basic XL nämlich nicht eingeben. Wenn Sie alle dazugehörigen Erklärungen befolgen, kann nichts schiefgehen.

Besondere Hinweise zu Turbo-Basic XL

Turbo-Basic XL ist als »AUTORUN.SYS«-File ausgelegt. Das heißt, wenn Sie den Interpreter komplett eingegeben haben, können Sie Ihrem File einfach den Namen »AUTORUN.SYS« geben. Nach dem Einschalten des Computers wird dann automatisch Turbo-Basic XL geladen. Anschließend können Sie auch schon mit dem Interpreter arbeiten. Einzige Voraussetzung: Die Files »DOS.SYS« und »DUP.SYS« (entweder DOS 2.0 oder 2.5) müssen sich auf der gleichen Diskette befinden. Mit DOS 3.0 arbeitet Turbo-Basic XL übrigens nicht!

Hier noch einige Hinweise, wie Sie einen Dateinamen auf Diskette ändern können.

Nehmen wir an, Sie haben Ihrem Interpreter bei der Eingabe mit AMPEL den Namen »TURBO« gegeben. Dieses File soll nun den Namen »AUTORUN.SYS« erhalten. Entfernen Sie also zunächst Ihre Turbo-Basic-XL-Diskette und schalten Sie bitte den Computer aus. Nachdem Sie eine formatierte Diskette, auf der sich die Files »DOS.SYS« und »DUP.SYS« befinden, ins Laufwerk gelegt haben, schalten Sie den Computer wieder ein. Anschließend geben Sie »DOS« ein und gelangen über die <RETURN>-Taste ins DOS-Menü. Nun rufen Sie auf der Turbo-Basic-XL-Diskette mit »E« die Funktion »Rename File« auf und drücken <RETURN>. Die Meldung »Give – Old Name, New Name« erscheint. Geben Sie beispielsweise »TURBO,AUTORUN.SYS« ein. Sollte Ihre Datei einen anderen Namen haben, müssen Sie diesen anstelle von »TURBO« einsetzen. Nach Betätigung der <RETURN>-Taste wird Ihr File umbenannt und hat dann den Namen »AUTORUN.SYS«. Überzeugen Sie sich, daß der soeben beschriebene Vorgang auch wirklich funktioniert hat. Mit »A« für Directory und <RETURN> wird das neue Directory auf dem Bildschirm angezeigt. Folgende Programme sollten sich jetzt auf Ihrer Diskette befinden: DOS.SYS, DUP.SYS und AUTORUN.SYS.

Noch eins, bevor Sie mit Turbo-Basic XL arbeiten: Sie sollten sich mindestens eine Sicherheitskopie Ihrer Diskette anlegen. Nehmen Sie sich also eine neue oder eine Diskette, die Sie löschen können, und formatieren Sie diese. Falls Sie über DOS 2.5 verfügen, können Sie sich das Formatieren sparen. Mit der Funktion »J« im DOS-Menü wird nun die Diskette kopiert. Wenn Sie ein Diskettenlaufwerk besitzen, geben Sie jetzt »D1,D1« ein und betätigen daraufhin die <RETURN>-Taste. Der Kopiervorgang beginnt. Dabei werden Sie mehrmals aufgefordert, die Original-Diskette mit der Duplikat-Diskette auszutauschen.

Sie haben jetzt also zwei Disketten vorliegen, die exakt die gleichen Programme enthalten. Arbeiten Sie aber bitte nur mit einer der Disketten und legen Sie die andere an eine sichere Stelle.

Nach dieser Sicherheitsmaßnahme können Sie endlich Turbo-Basic XL ausprobieren. Dazu muß sich nur Ihre Interpreter-Diskette im Laufwerk befinden. Schalten Sie jetzt die Stromversorgung Ihres Computers aus und wieder ein. Wenn Sie dabei keine Taste, auch nicht die

```
250 END
999 ------------------------------------
1000 PROC RECHNEN
1010   GRAPHICS 15+16:POKE 764,255
1030   DX=(XMAX-XMIN)/159
1040   DY=(YMAX-YMIN)/191
1050   CX=XMIN:CY=YMAX
1080   FOR ZEILE=0 TO 191
1090     FOR SPALTE=0 TO 159
1110       TI=0:XW=0:YW=0:X2=0:Y2=0
1120       WHILE TI<TMAX AND (X2+Y2)<8
1130         YW=2*XW*YW-CY:XW=X2-Y2-CX
1150         X2=XW^2:Y2=YW^2:TI=TI+1
1180       WEND
1190       IF TI=TMAX
1200         COLOR %0
1220       ELSE
1230         COLOR (TI MOD 3)+%1
1240       ENDIF
1250       CX=CX+DX:PLOT SPALTE,ZEILE
1260       IF PEEK(764)=28 THEN POP :GOT
O 1295
1270     NEXT SPALTE
1280     CX=XMIN:CY
```

Ein typischer Turbo-Basic-XL-Bildschirm



<OPTION>-Taste, mit der normalerweise das eingebaute Basic abgestellt wird, betätigt haben, erscheint das Titelbild und kurz danach die READY-Meldung. Sie befinden sich jetzt in Turbo-Basic XL. Viel Spaß beim Programmieren!

Strukturierte Programmierung

In der folgenden Beschreibung bedeuten:
aexp=arithmetic expression (arithmetischer Ausdruck)
sexp=string expression (Text Ausdruck (A$, "TEXT", CHR$(), STR$(), HEX$())
lineno=Zeilennummer
»...«=Ein oder mehrere Befehle. In einer IF-THEN-Abfrage handelt es sich hierbei um den nach THEN folgenden Text. Sonst gibt es keine Beschränkung. »...« kann also auch mehrere Programmzeilen und/oder den Teil einer Programmzeile umfassen.

Es gibt im Standard-Atari-Basic nur zeilennummerorientierte Sprungbefehle und die »FOR...NEXT«-Schleife zur Konstruktion von Schleifen. In Turbo-Basic gibt es zusätzliche, an Pascal angelehnte Strukturelemente.
IF aexp THEN lineno
IF aexp THEN ...
Das normale IF...THEN Statement.

IF aexp ... **ENDIF**
IF aexp ... **ELSE** ... **ENDIF**

Wenn die Bedingung aexp erfüllt (<>0) ist, wird der Programmteil zwischen IF und ELSE, sonst der zwischen ELSE und ENDIF ausgeführt. ELSE kann auch entfallen. Als Trennzeichen nach aexp dient nicht THEN, sondern ein Doppelpunkt (:) oder Zeilenende (RETURN). Auch vor und nach ELSE und ENDIF muß ein solches Trennzeichen stehen. Nach ELSE darf keine Zeilennummer folgen. In Atari-Basic benötigt man dafür oft mehrere GOTO-Befehle.

REPEAT ... **UNTIL** aexp

Wiederholt die Anweisungen »...«, bis die Bedingung erfüllt ist. Hier wird die Bedingung erst am Ende der Schleife geprüft. Der Programmteil zwischen REPEAT und UNTIL wird also mindestens einmal ausgeführt.

WHILE aexp ... **WEND**

Wiederholt »...«, solange die Bedingung erfüllt ist, das heißt, wenn die Bedingung das erste Mal nicht erfüllt ist, wird die Schleife kein einziges Mal ausgeführt.

DO ... LOOP

Endlosschleife. Wiederholt die Anweisungen »...« immer wieder von neuem.

EXIT

Verläßt eine Schleife, Sprung ans Schleifenende. Dieser Befehl ist verwendbar bei »DO ... LOOP« und »REPEAT ... UNTIL«, »WHILE ... WEND« und auch bei »FOR ... NEXT«-Schleifen. Dieser Befehl stellt eine Art Notausgang aus Schleifen dar, bei »DO ... LOOP« sogar den einzigen, der in strukturierten Programmen erlaubt ist. Es wird stets an das Ende der Schleife gesprungen. Eine Schleife läßt sich auch mit »POP:GOTO lineno« abbrechen. Dies sollte jedoch nur im Notfall geschehen, da dieser Befehl nicht zur übersichtlichen Programmierung beiträgt.

★F
★F +

Nach diesem Befehl sind »FOR ... NEXT«-Schleifen abweisend. Es wird also vor dem ersten Durchlauf der Schleife geprüft, ob der Zähler schon den Endwert erreicht hat. Dazu ein Beispiel: »FOR I=2 TO 1:? I:NEXT I«.
1. Der Anfangswert der Schleife ist 2
2. Schleifenzähler um 1 erhöhen
3. Vergleich mit dem Endwert (1)
4. Verlassen der Schleife, da die Bedingung erfüllt ist.

Nach ★F + wird zuerst die Variable I mit dem Anfangswert 2 geladen und dann mit der Variablen I (Endwert=1) verglichen. Anschließend wird die Schleife bis zum »NEXT I«-Befehl übersprungen. Es erfolgt keine Ausgabe auf dem Bildschirm. Der »F+«-Befehl erspart oft eine spezielle Abfrageroutine.

★F -

Stellt den Normalzustand wieder her. »FOR ... NEXT«-Schleifen werden mindestens einmal durchlaufen. Auch bei RUN wird automatisch ein »★F -« ausgeführt (entspricht dem Atari-Basic).

PROC name

Beginn eines Unterprogramms (Prozedur) mit dem Namen name.

ENDPROC

Prozedurende. Entspricht dem RETURN nach einem »GOSUB lineno«.

EXEC name

Ruft die Prozedur name auf, entspricht »GOSUB lineno«. Die normalen Befehle GOSUB und RETURN erlauben, nur Unterprogramme mit einer Zeilennummer aufzurufen. Hier ist der Aufruf mit einem leichter zu behaltenden Namen möglich. Außerdem belegt EXEC-PROC-ENDPROC meist sogar weniger Speicherplatz als GOSUB-RETURN und ist etwas schneller. Die Prozedurnamen werden genauso gespeichert wie die Variablennamen. Jeder Name belegt 8 Byte plus ein Byte für jedes Zeichen. Jede weitere Verwendung benötigt dann nur ein (zwei) Byte. Die Zeilennummer einer GOSUB-Anweisung belegt dagegen 7 Byte. Obwohl Turbo-Basic XL kompatibel zu Atari-Basic ist, können statt 128 jetzt 256 verschiedene Variablen- oder Prozedurnamen verwendet werden. Ab der 129sten Variablen kostet jede Verwendung zwei statt bisher nur 1 Byte Speicherplatz.

ON aexp **EXEC** pname,pnam...

Entspricht der Atari-Basic-Anweisung »ON GOSUB«. pnam=PROCedurname.

\# name
GO# name
ON aexp **GO#** nam,nam,nam
TRAP #name
RESTORE # name

Labeldefinition, das # entspricht dem PROC. In Turbo-Basic XL läßt sich der unstrukturierte Sprungbefehl GOTO wenigstens dadurch lesbarer machen, daß »GOTO zeilennummer« durch »GO# name« ersetzt wird. Auch bei TRAP und RESTORE können Marken verwendet werden. Dabei wird die Kennung »#« verwendet, um dem Interpreter mitzuteilen, daß keine Zeilennummer, sondern ein Name folgt. Übrigens ist »GO# name« schneller als »GOTO zeile«. Die Marken für GO#, TRAP# und RESTORE# werden mit # gekennzeichnet, die für EXEC mit PROC. Diese strikte Trennung dient zur Erhöhung der Gliederung und Transparenz von Programmen.

POP

POP gilt für Unterprogramme mit GOSUB und EXEC ebenso wie für die Schleifen »FOR ... NEXT«, »REPEAT ... UNTIL«, »WHILE ... ENDWHILE« und »LOOP ... ENDLOOP«. Auf dem Runtimestack belegen GOSUB, EXEC, REPEAT, WHILE und LOOP jeweils 4 Byte (wie GOSUB in Atari-Basic). »FOR ... NEXT« belegt jedoch 13 statt 12 Byte, da jetzt 256 verschiedene Variablen zur Verfügung stehen. Auf dem Stack werden jetzt aber nicht mehr die Zeilennummern gespeichert, ein Grund für das langsame Atari-Basic, sondern die Speicheradressen dieser Zeilen. Dadurch werden Schleifen schneller. Außerdem sinkt die Geschwindigkeit der Schleifenwiederholung nicht mit steigender Entfernung vom Programmanfang! Trotzdem konnte einer der Vorzüge von Atari-Basic beibehalten werden: Ein Programm, das beispielsweise durch einen Programmfehler angehalten wurde, kann man editieren und dann mit CONT oder GOTO fortsetzen. Dabei werden weder die Variablen noch der Stack gelöscht. So lassen sich auch Programme schreiben, die ihre DATA-Zeilen selbst generieren oder nicht mehr benötigte Programmteile löschen (mit POKE 842,13, danach verhält sich der Computer so, als würde ständig RETURN gedrückt, bis POKE 842,12 den Normalzustand wieder herstellt). Zum Löschen von Programmteilen gibt es in Turbo-Basic XL den DEL-Befehl.

\- -

Dies ist eine spezielle REM-Anweisung. Bei der Eingabe einer Programmzeile wird alles nach »- -« nicht beachtet. Bei LIST werden aber nicht 2, sondern 30 Minuszeichen gedruckt. Diese Anweisung belegt sogar ein Byte weniger Speicherplatz als ein REM ohne Text. Eine entsprechende REM-Zeile im normalen Basic benötigt wesentlich mehr Speicherplatz.

LIST

Beim LISTen von Programmen werden Schleifen optisch durch Einrücken um jeweils zwei Leerzeichen hervorgehoben. Programme sind so übersichtlicher und strukturierter. Außerdem lassen sich auf diese Art und Weise manche Fehler vermeiden. Bei unsauberer Programmierung (mehrere NEXT zu einem FOR, mehrere ENDPROCs zu einem PROC...) gerät das Listing, ebenso wie der Interpreter, allerdings in Unordnung. Solche Konstruktionen, die in Atari-Basic und vielen anderen Basic-Versionen nur schwer zu umschreiben sind, können in Turbo-Basic XL leicht durch IF-ELSE-ENDIF oder EXIT ersetzt werden. Daraus resultieren wieder leichter lesbare Programme.

★L -

schaltet die Tabulierung ab. Dies kann notwendig sein, um lange Programmzeilen zu editieren oder um Platz beim Speichern auf Diskette (LIST"D:X") zu sparen. Die Trennzeile »- -« wird nach »★L -« nur als doppeltes statt dreißigfaches Minus-Zeichen gelistet.

★L
★L +

schaltet die Tabulierung wieder ein (Normalzustand nach dem Laden des Interpreters). Übrigens gibt es jetzt auch ein LIST von einer Programmzeile bis zum Programmende. »LIST 3000,« listet ab Zeile 3000 bis zum Programmende, »LIST"P:",3000,« entsprechend auf den Drucker (das einzelne Komma »,« nach der Zeilennummer veranlaßt den Computer, automatisch 32767 für die höchstmögliche Zeilennummer zu ergänzen).

Neue Fehlermeldungen

Es gibt folgende neue Fehlermeldungen:

ERROR - 22 ?NEST

Schachtelungsfehler, tritt auf, wenn das zu einem WHILE gehörende ENDWHILE nicht gefunden wird, oder das ENDIF zu einem IF, oder auch, nach »★F +«, das NEXT zu einem FOR. Beim Verlassen von Unterprogrammen (durch RETURN oder ENDPROC) werden Schleifen abgebrochen, dies gilt sowohl – wie gewohnt – für die »FOR ... NEXT«-Schleife, wie auch für die anderen unter Turbo-Basic XL zur Verfügung stehenden Schleifen.

ERROR - 16 ?GOSUB

Zu einem GOSUB fehlt RETURN.

ERROR - 13 ?FOR

Zu einem NEXT fehlt FOR.

ERROR - 23 ?WHILE

Zu einem WEND fehlt WHILE.

ERROR - 24 ?REPEAT

Zu einem UNTIL fehlt REPEAT.

ERROR - 25 ?DO
Zu einem LOOP fehlt DO.
ERROR - 28 ?EXEC
Zu einem ENDPROC fehlt EXEC.
ERROR - 29 ?PROC
Eine unbekannte Prozedur wurde aufgerufen.
ERROR - 30 ?#
Eine unbekannte Marke wurde verwendet.
ERROR - 27 XPROC
(Executing PROC). Diese Fehlermeldung tritt auf, wenn eine PROC-Anweisung ausgeführt wird. Prozeduren dürfen nur von EXEC aufgerufen werden.
ERROR - 26 ?EXIT
EXIT ohne Schleife.
ERROR - 15 ?DEL
Das GOSUB zu einem RETURN, NEXT zu einem FOR, REPEAT zu einem UNTIL ... wurde gelöscht. In Atari- und in Turbo-Basic XL lassen sich Programme editieren, ohne Variablenwerte oder den Stapel zu zerstören. Wenn dann bei Rückkehr aus einem Unterprogramm (Schleife) die entsprechende Zeile gelöscht oder verändert wurde, kann dieser Fehler passieren. Dies tritt auch auf, wenn ein in ein Programm eingebautes DEL sich selbst löscht. Alle Fehlernummern, die Atari-Basic ohne irgendwelche Texte ausgibt, werden in Turbo-Basic XL grundsätzlich mit einem kurzen Text ergänzt (beispielsweise »138 TIMEOUT«, »29 ?PROC« etc.). Ausführlichere Erläuterungen können dem Atari-Basic Referenz Manual oder der DOS-Anleitung entnommen werden. Längere Texte würden, bei den insgesamt 60 zur Verfügung stehenden Fehlern, den Platzbedarf des Interpreters noch wesentlich erhöhen.
DEL von,bis
Löscht die Programmzeilen von - bis (jeweils einschließlich).
RENUM alt,neu,incr
Numeriert alle Programmzeilen ab Zeile alt um. Die neuen Zeilennummern beginnen bei neu und werden jeweils um incr erhöht. Alle Zeilennummern vor alt bleiben unverändert. Dieser Befehl ändert auch die Zeilennummern nach GOTO, GOSUB, TRAP, RESTORE, LIST, DEL, ON-GOTO und ON-GOSUB. Bei Verwendung undefinierter Zeilennummern wird die entsprechende negative Zahl eingesetzt (beispielsweise GOTO-100). Bei berechneten Sprüngen (GOTO VAR, GOSUB 100+10★A, RESTORE A★10+1000) versagt RENUM. Wenn hinter dem Befehl eine Zahl folgt »GOTO 1000+A★10«, wird die Zahl wie bei einem normalen Befehl »GOTO 1000« behandelt. Der nachfolgende Programmteil der Zeile bleibt jedoch unverändert. Wenn keine Zahl, sondern ein Variablenname oder eine Klammer folgt, wird der Befehl überhaupt nicht verändert.

DUMP
DUMP filespec
Dieser Befehl erzeugt eine Liste der verwendeten Variablen. Wie bei LIST kann die Ausgabe auch auf einem Drucker erfolgen (DUMP "P:").
Ein Beispiel:

A = 100	numerische Variable
B(10,1	Array, DIM B(9) oder DIM B(9,0)
C(0,0	undimensioniertes Array. Bei Arrays werden die beiden möglichen Dimensionen stets um 1 erhöht angezeigt.
D(10,10	DIM D(9,9)
E$ 10,20	String, LEN=10, DIM E$(20)
F$ 0,0	nicht dimensionierter String
G$ 0,10	DIM G$(10), LEN(G$)=0
H PROC 100	PROC H in Zeile 100
I # 120	Marke I in Zeile 120
J ?	Undefinierte Marke oder PROC

Die Ausgabe der Variablen/Marken erfolgt in der Reihenfolge, wie sie in der Variablentabelle gespeichert sind.
TRACE
TRACE +
Schaltet den TRACE-Modus ein. Das heißt, die Nummer jeder ausgeführten Zeile wird in eckigen Klammern auf dem Bildschirm ausgedruckt.
TRACE -
Hebt TRACE wieder auf. Der TRACE-Modus wird außerdem aufgehoben, sobald eine Fehlermeldung auftritt. Die Zeilennummern eines PROC oder # werden bei Aufruf mit EXEC oder GO# nicht ausgegeben.
★B
★B +
Nach diesem Befehl wird das Drücken der <BREAK>-Taste wie jeder andere Fehler behandelt. Eine Programmunterbrechung läßt sich mit TRAP abfangen und so vor einer versehentlichen Unterbrechung schützen.
★B -
Hebt den oben erwähnten Modus wieder auf. Bei RUN wird »★B -« automatisch ausgeführt.

Befehle

Erläuterung:
<=> steht für: entspricht im normalen Basic
DPOKE adr,word
Doppel-Byte-POKE <=> POKE adr,word-256★INT(wort/256):POKE adr+1,INT(wort/256)
MOVE source,dest,count
Blocktransfer <=> FOR I=0 TO COUNT-1:POKE dest+I,PEEK(source+I):NEXT I
Mit »MOVE 57344,NEUCHARSET,1024« läßt sich beispielsweise der Zeichensatz kopieren.
-MOVE source,dest,count
Blocktransfer, zuerst wird das letzte Byte verschoben. Dies ermöglicht, einen Speicherbereich zu einer höheren Adresse zu verschieben, ohne daß es bei Überlappungen (wenn source+count>dest) zu Zerstörungen kommt. <=> FOR I=COUNT-1 TO 0 STEP -1:POKE dest+I,PEEK (source+I):NEXT I. MOVE kann auch benutzt werden, um einen Speicherbereich zu füllen. Beispiel: »POKE DPEEK(88),128:MOVE DPEEK(88),DPEEK(88)+1,959«. Dies schreibt den Textbildschirm mit dem Bildschirmcode für das inverse Leerzeichen voll. Obwohl hier viele, eigentlich unnötige Ladevorgänge stattfinden, ist dies viel schneller als eine Basic-Schleife.
BPUT #n,adr,len
Blockschreiben <=> FOR I=0 TO len-1:PUT #n,PEEK (adr+I):NEXT I
BGET #n,adr,len
Blocklesen <=> FOR I=0 TO len-1:GET #n,A:POKE adr+I,A:NEXT I
Mit diesen Befehlen kann man Speicherbereiche mit maximaler Geschwindigkeit speichern und laden. Beispiel:
OPEN #1,8,0,"D:BILD.PIC":BPUT #1,DPEEK(88),7680: CLOSE #1
OPEN #1,4,0,"D:BILD.PIC":BGET #1,DPEEK(88),7680: CLOSE #1
Speichern oder Laden eines Grafik-8-(oder 9,10,11,15)-Bildes auf Diskette. Achtung: Die Zahl 7680 muß je nach Grafikmodus geändert werden (sonst werden nicht mehr zum Bildspeicher gehörende Speicherbereiche mitgelesen oder -geschrieben)
%PUT
Mit diesem Befehl lassen sich Zahlen schneller und kompakter auf Diskette oder in einer RAM-Disk speichern. Dabei werden immer sechs Byte gepeichert.



%GET
Zum Lesen von Zahlen, die mit %PUT auf ein Speichermedium gespeichert wurden. Ein Beispiel für die Anwendung von %PUT und %GET finden Sie auf Seite 70.

FILLTO x,y
Kurzschreibweise, schneller und übersichtlicher als: »POSITION x,y:XIO 18,#6,0,0,"S:"FCOLOR n«
Wählen der Farbe für FILLTO. In Standard-Basic heißt dieser Befehl »POKE 765,n«.

CLS
CLS #6
Bildschirmlöschen. CLS < = > A=PEEK(766):POKE 766,0:POSITION 0,0:? (#6;)CHR$(125);:POKE 766,A

PUT n
< = > ?CHR$(n); beispielsweise »PUT 253« für »? CHR$(253);«. Bei PUT, GET, INPUT... ist auch die Angabe »#0« möglich. Die Benutzung von »IOCB 0« (mit »CLOSE« oder »OPEN #0«) verhindert allerdings das ordnungsgemäße Arbeiten des Bildschirm-Editors, gegebenenfalls <SYSTEM-RESET> drücken. Fehlt bei PUT die #-Angabe, so wird automatisch »IOCB #0« benutzt.

GET KEY
< = > OPEN #7,4,0,"K:":GET #7,KEY:CLOSE #7. Wartet auf Tastendruck und weist der Variablen KEY den ATASCII-Wert der gedrückten Taste zu (der Variablenname ist frei wählbar).

DIM
Beim DIM-Befehl werden Arrays und Strings automatisch gelöscht, also auf 0 gesetzt. DIM A(100) < = > DIM A(100):FOR I=0 TO 100:A(I)=0:NEXT I

INPUT "text",var,var...
INPUT "text";var,var...
INPUT ähnelt jetzt sehr stark dem entsprechenden INPUT-Befehl unter Microsoft-Basic. Ein Text nach INPUT erspart die sonst nötigen PRINT-Befehle. Folgt nach dem Text ein Semikolon statt eines Kommas, so wird zusätzlich ein ? ausgegeben. Mit »INPUT " ";A« erreicht man ein Input ohne das manchmal störende »?«.

TEXT x,y,sexp
TEXT x,y,exp
Schreibt einen Text in ein Grafik-Bild. »x,y« stellt die Position der oberen linken Ecke des ersten Zeichens des String-Ausdrucks (wird in Bildpunkten gezählt) dar. Beispiel: »GRAPHICS 8:TEXT 50,90,"Turbo-Basic":TEXT 70,95,1000«. Im Gegensatz zum normalen PRINT-Befehl darf nach TEXT nur ein Ausdruck erfolgen (keine Liste mit Komma oder Semikolon). Außerdem werden Texte am Zeilenende abgebrochen, es gibt auch kein Scrolling.

CIRCLE x0,y0,r
CIRCLE x0,y0,xr,yr
Zeichnet einen Kreis um den Punkt x0,y0 mit dem Radius r. Bei Angabe zweier (unterschiedlicher) Radien für die x- und y-Richtung entstehen Ellipsen.

PAINT x,y
Füllt eine geschlossene Figur, beispielsweise einen Kreis, mit der mit COLOR gewählten Farbe. Dieser Befehl kann fast jede beliebige Figur mit einer bestimmten Farbe füllen. Da hierbei eine für den 6502 angepaßte, rekursive Funktion benutzt wird, ist der Komplexität der auszufüllenden Figur durch den freien Speicherplatz eine Grenze gesetzt. Im Extremfall würde ein Arbeitsbereich von etwa 90 KByte RAM benötigt. Auch einfache und kleinere Figuren belegen vorübergehend einige hundert Byte. Sollte der Speicher überlaufen (FRE(0) zu klein), so gibt es die Fehlermeldung »ERROR - 2 MEM«. Die PAINT- und die TEXT-Routine benutzen eigene, schnelle Plot-Routinen, da hierbei viele nebeneinander liegende Punkte relativ einfach berechnet werden können. Hiervon machen die Routinen im Atari-Betriebssystem leider keinen Gebrauch.

TIME$= siehe unten

PAUSE n
Unterbricht die Programmausführung für n/50-Sekunden. PAUSE ersetzt das ungenaue und speicherplatzfressende Timing mit leeren FOR-NEXT-Schleifen. In Atari-Basic wird oft die Potenzierung (A=1^1) für eine kleine Verzögerung benutzt. Da Turbo-Basic XL bei der Potenzierung besonders schnell ist, sollte dafür beispielsweise »PAUSE 9« benutzt werden.

DSOUND voice,freq,dis,vol
Ähnlich dem normalen SOUND-Befehl. Der Atari-Computer kann zwei normale Stimmen zu einer zusammenfassen. Die Frequenzauflösung beträgt dann 16 Bit (0..65535) statt 8 Bit (0..255). Die resultierende Frequenz berechnet sich (in Hertz) zu: 1789790/(2★freq+14) statt sonst 63921/(2★freq+2). Diese Werte stammen aus einer amerikanischen Veröffentlichung und unterscheiden sich eventuell etwas von den Werten bei den deutschen Atari-Versionen, da sie von der Videofrequenz abgeleitet sind, die in den USA einen anderen Wert besitzt (siehe auch Seite 124).

DSOUND
Kurzform für: FOR I=0 TO 3:SOUND I,0,0,0:NEXT I

CLOSE
Kurzform für: FOR I=1 TO 7:CLOSE #I:NEXT I

Funktionen

DPEEK(adr)
Doppel-Byte-PEEK < = > PEEK(adr)+256★PEEK(adr+1)

INKEY$
Spezialvariable. Wenn eine Taste gedrückt wird, enthält INKEY$ das entsprechende Zeichen. Wenn nicht, enthält sie einen Leerstring (" "). So läßt sich ein Tastendruck verarbeiten, ohne den Programmablauf zu unterbrechen.

INSTR(A$,B$)
INSTR(A$,B$,i)
Sucht einen String B$ in einem (längeren) String A$. Wenn gefunden, wird die Position von B$ in A$ zurückgeliefert, sonst eine Null. »i« stellt den Index (Position) dar, ab dem die Suche beginnen soll.

UINSTR(A$,B$)
UINSTR(A$,B$,i)
Ähnlich INSTR. Die Bits 7 und 5 der einzelnen Zeichen werden nicht beachtet. Bei der Suche nach »MODEM« läßt sich so auch »Modem«, »MoDeM« oder der entsprechende inverse Text finden (UINSTR steht für UppercaseINSTR). Als Nebeneffekt werden bei der Suche nach Zahlen oder Satzzeichen auch die Spezial-Zeichen gefunden ("!"=CTRL-A, "0"=CTRL-P etc.).

ERR
Kurzform für »PEEK(195)« zur Ermittlung des Fehlercodes.

ERL
Kurzform für »PEEK(186)+256★PEEK(187)« oder »DPEEK (186)« zur Ermittlung der Zeile, in der ein Fehler aufgetreten ist. ERR und ERL sollten in TRAP-Routinen verwendet werden.

TIME
Spezialvariable, enthält die Zeit (vom internen Timer RTCLOCK des Atari-Computers, ausschlaggebend sind die Speicherzellen 18 bis 20) in 1/50-Sekunden.

TIME$
Spezialvariable, enthält die Zeit als sechsstelligen String im Format hhmmss (hh=Stunde 00 bis 23, mm=Minute 00 bis 59, ss=Sekunde 00 bis 59).

TIME$=
Zum Stellen der Uhr. »TIME$= "151520"« stellt die Uhr auf 15 Uhr, 15 Minuten und 20 Sekunden. Die Variable TIME läßt sich nicht direkt verändern. Statt dessen wird entweder »TIME$=« verwendet oder entsprechend in die Speicherstellen 18 bis 20 gePOKEt. Die Uhr geht nicht ganz genau, da die Frequenz, mit der im Atari die Fernsehbilder erzeugt werden, nicht genau 50 Hz beträgt. TIME$ wird von dieser Frequenz abgeleitet.

FRAC(exp)
Diese Funktion ermittelt den Nachkommaanteil einer Zahl. »FRAC(exp)« ist nicht immer gleich »exp-INT(exp)«, da INT die nächstkleinere Zahl ermittelt. So ergibt »? INT(-0.3)« -1, »? FRAC(-0.3)« ergibt -0.3.

TRUNC(exp)
Diese Funktion ermittelt den ganzzahligen Anteil einer Zahl. Dies ist die zu FRAC komplementäre Funktion. »? TRUNC(-0.3)« ergibt 0.

RND
RND (irgendwas) kann in Turbo-Basic XL abgekürzt werden, indem man die Klammern wegläßt. Es wird also RND statt RND(0) verwendet (speicherplatzsparend).

RAND(n)
Dies ist die Kurzschreibweise für »TRUNC(RND(0)★n)« und erzeugt eine ganze Zufallszahl zwischen einschließlich 0 und (ausschließlich) n.

HEX$(exp)
Ähnlich STR$. HEX$ wandelt die Integerzahl exp (0<=exp<=65535) in einen sedezimalen (hexadezimalen) String um. Wenn exp kleiner als 256 ist, ergibt dies einen zweistelligen String, sonst einen vierstelligen.

DEC(sexp)
Ähnlich VAL, Umkehrung zu HEX$. Der String sexp wird in eine dezimale Integerzahl gewandelt. Wenn sexp mehr als vier gültige sedezimale Ziffern enthält, so gelten nur die letzten vier.

$aaaa
Sedezimalzahl im Programm. Beispiel:
FOR I=$0600 TO $067F:READ A:POKE I,A:NEXT I
statt:
FOR I=1536 TO 1663:READ A:POKE I,A:NEXT I

& Binäres AND
! Binäres OR
EXOR Binäres exclusiv OR
Diese drei Operatoren arbeiten mit 16-Bit-Integern. Also mit Zahlen zwischen 0 und 65535 und nicht mit Boolschen Größen (1 oder 0) wie die Operatoren AND, OR und NOT.

DIV
Division ohne Rest: a DIV b <=> TRUNC(a/b)

MOD
Bestimmung des Divisionsrestes (Modulo): a MOD b <=> a-b*TRUNC(a/b)

%0 %1 %2 %3
Die Zahlen 0 bis 3 sind als Konstanten definiert. Die Verwendung einer Zahl (auch $aaaa) im Programm kostet jedesmal 7 Byte, die Verwendung einer Variablen, unabhängig von der Länge des Namens, jedoch nur 1 oder 2 Byte. Auch die Verwendung von %0 bis %3 belegt ebenfalls jeweils 1 Byte, aber keinen Eintrag in der auf 256 Variablen begrenzten Variablentabelle.

In Strings oder in zwischen Anführungszeichen stehendem Text ist es jetzt möglich, »"« (entspricht CHR$(34)) durch doppelte Anführungszeichen ("") einzufügen. Beispiel: »? "TEST" "TEXT"« erzeugt den Ausdruck: »TEST"TEXT«. In Atari-Basic muß man dafür schreiben: »? "TEST";CHR$(34);
"TEXT"«. Bei Zuweisungen (A$= "TEST" "TEXT") ist das (in Atari-Basic) noch etwas umständlicher.

Turbo-Basic XL wandelt bei der Eingabe von Programmzeilen automatisch kleine in große und inverse in normale Buchstaben um. Dies gilt natürlich nicht zwischen Anführungszeichen (") sowie nach REM oder DATA-Befehlen. Somit lassen sich Programme auch mit Kleinbuchstaben eingeben, ohne ständig zwischen Groß- und Kleinbuchstaben (mit der <CAPS>-Taste) hin- und herzuschalten oder ständig <SHIFT> drücken zu müssen.

In Variablen- und Prozedurnamen ist außer den Buchstaben und Ziffern auch das Unterstreichungszeichen »_« (<SHIFT ->) zugelassen. So sind endlich Namen wie MAX_LEN oder PROC SORT_KUNDEN erlaubt.

Diskettenbefehle

DIR
DIR "D1:★.★"
Ausgabe des Inhaltsverzeichnisses einer Diskette auf dem Bildschirm. »DIR "D1:★.★"« listet alle Files von Diskette 1, »D2:A★.★« alle Files von Diskette 2, deren Name mit A beginnt. »D1:★.★« kann entfallen, dann wird automatisch »D:★.★« eingesetzt (ähnlich wie im DOS-Menü mit »A RETURN RETURN«). Bei diesem Befehl sind Stringvariablen wie auch bei den folgenden Befehlen zulässig.

RENAME "D:OLD,NEW"
Benennt das File OLD in NEW um. Entspricht: »XIO 32,#7,0,0,"D2:OLD,NEW"« oder der DOS-Menü Auswahl E.

DELETE "D:FILE"
Löscht ein File (entspricht »XIO 33« oder der DOS-Menü Auswahl D).

LOCK "D:FILE"
Schützt ein File gegen Überschreiben (entspricht »XIO 35« oder der DOS-Menü Auswahl F).

UNLOCK "D:FILE"
Hebt den Schreibschutz wieder auf (entspricht »XIO 36« oder der DOS-Menü Auswahl G).

BLOAD "D:FILE.OBJ"
Lädt ein binäres File (entspricht der DOS-Menü Auswahl L mit »/N« nach dem Filenamen).

BRUN "D:FILE.OBJ"
Lädt ein binäres File und startet es, wenn eine RUN-Adresse im File enthalten ist (entspricht der DOS-Menü Auswahl L, ohne angehängtes »/N«).

Damit sind die zusätzlichen Befehle und Funktionen von Turbo-Basic XL erklärt. Selbstverständlich stehen darüber hinaus noch alle anderen Befehle, die man vom normalen Basic her kennt, zur Verfügung. Mit Turbo-Basic XL wird die Programmierung auf dem Atari 800XL/130XE zur wahren Freude. Befehle, die man sich schon lange gewünscht hat, stehen jetzt endlich zur Verfügung.

Autostart
Mit Turbo-Basic XL ist es möglich, nach dem Einschalten des Computers ein Basic-Programm automatisch zu laden und zu starten. Dazu benennen Sie Ihr Basic-Programm einfach in AUTORUN.BAS um. Vergessen Sie nicht: Das eigentliche Turbo-Basic muß dann mit dem Namen AUTORUN.SYS auf Diskette vorliegen.

(Frank Ostrowski/Werner Breuer/hf)

Steckbrief	
Programm:	Turbo-Basic XL
Sprache:	Assembler
Eingabehilfe:	AMPEL
Datenträger:	Diskette, Kassette

```
0000:FF FF E7 02 E8 02 29 36<88>
0008:00 5E 87 5E 00 18 18 FF<E5>
0010:3C 3C 66 42 00 00 00 06<14>
0018:66 7E 76 66 00 00 18 3C<7C>
0020:66 7E 66 66 00 3E 73 33<28>
0028:3E 30 30 78 00 7C 37 33<64>
0030:3E 30 30 78 00 00 60 63<EC>
0038:36 3C 39 63 00 01 01 0F<D4>
0040:03 03 F6 E4 00 80 80 F0<7C>
0048:C0 C0 60 20 00 01 03 06<16>
0050:06 06 03 01 00 E1 33 06<35>
0058:06 06 33 E1 00 C7 63 33<46>
0060:36 36 66 CE 00 0E 9C 9C<7F>
0068:F6 66 66 67 00 7E 33 33<A3>
0070:3E 30 30 78 00 67 66 66<58>
0078:66 66 66 3C 00 FF 31 31<A4>
0080:31 31 31 7B 00 FB 81 81<A1>
0088:F1 81 81 FB 00 F8 8C 8C<C4>
0090:F8 B0 98 CC 00 21 93 21<33>
0098:A2 00 BD 00 E0 9D 00 5C<52>
00A0:BD 00 E1 9D 00 5D BD 00<54>
00A8:E3 9D 00 5F E8 D0 EB A9<34>
00B0:10 8D C6 02 A9 5C 8D C5<49>
00B8:02 8D F0 02 8D F4 02 A9<E4>
00C0:45 8D 44 03 A9 21 8D 45<67>
00C8:03 A9 4F 8D 48 03 8E 49<11>
00D0:03 A9 0B 8D 42 03 20 56<93>
00D8:E4 8E F0 02 60 7D 9B 7F<35>
00E0:7F 20 00 01 02 03 04 05<61>
00E8:06 07 9B 7F 7F 08 09 0A<31>
00F0:0B 0C 0D 0E 0F 10 9B 9B<86>
00F8:7F 20 20 20 20 54 55 52<50>
0100:42 4F 2D 42 41 53 49 43<D9>
0108:20 58 4C 20 31 2E 35 9B<01>
0110:7F 20 28 63 29 20 31 39<6A>
0118:38 35 20 46 72 61 6E 6B<08>
0120:20 4F 73 74 72 6F 77 73<2A>
0128:6B 69 9B 9B E2 02 E3 02<16>
0130:00 21 80 20 A1 20 A9 80<C3>
0138:A0 20 85 0C 84 0D A9 FF<0C>
0140:8D 01 D3 20 00 00 A9 FE<74>
0148:8D 01 D3 A5 80 A4 81 8D<90>
0150:E7 02 8C E8 02 4C 5D E6<36>
0158:00 22 C9 22 A3 24 A3 24<4D>
0160:0B CB 28 C2 1D F1 B4 C2<93>
0168:00 E4 70 F5 BF F3 8B F4<95>
0170:9D FF 9D FF 9A FF FC F5<D9>
0178:11 25 E0 F5 A0 C8 4A C5<90>
0180:5B F5 57 DC A0 C8 6A F5<FC>
0188:65 E6 CC C3 E1 C2 43 C3<C4>
0190:0C C4 18 C4 2F C4 C8 C3<A3>
0198:04 F6 44 25 F2 DC 5A DC<66>
01A0:75 CA 62 CA 18 F8 32 F5<B2>
01A8:B0 F5 D3 F7 F2 DC 67 C4<5B>
01B0:54 C4 53 C2 74 C2 1B C2<F7>
01B8:FE 24 38 C2 AE C1 95 C4<05>
01C0:D9 C1 97 DD 8A C3 92 C3<49>
01C8:00 E4 DA F8 32 25 00 D8<94>
01D0:0A D8 FF F6 1E F4 9B F7<FA>
01D8:84 F7 B9 F7 8D F5 9A F5<49>
01E0:DA C5 DD C5 35 C2 1E F4<E9>
01E8:BB F8 52 F8 A0 C5 C6 C5<7A>
01F0:C9 C5 C3 C5 CC C5 A2 DC<CF>
01F8:74 FB B3 F8 82 F8 64 F8<FC>
0200:2E C2 92 F3 A3 24 E4 33<AD>
0208:00 C0 40 C0 58 FF 37 FC<49>
0210:74 35 77 35 AF F8 A3 24<7C>
0218:9B F5 D0 FD 3F FF D9 C1<59>
0220:80 DE 18 C6 1B C6 00 23<52>
0228:15 23 EA 20 24 23 4C 2B<2C>
0230:E5 44 3A 2A 2E 2A 9B 53<33>
0238:3A 9B 43 3A 9B 50 3A 9B<31>
0240:24 23 2E 23 AD 01 D3 29<2F>
0248:FC 09 02 8D 01 D3 60 3A<E3>
0250:23 28 36 45 D8 4E D8 65<17>
0258:D8 55 D8 5C D8 6C D8 8F<0D>
0260:DB 26 D8 14 D8 1D D8 2F<A9>
0268:D8 95 D8 84 D8 73 D8 A3<D4>
0270:24 89 C6 4C C6 1C C8 45<59>
0278:D8 4E D8 65 D8 55 D8 5C<6D>
0280:D8 6C D8 A3 24 38 D8 A2<44>
0288:C7 94 C6 90 C6 89 C6 90<76>
0290:C6 87 C6 34 DA 63 DA 59<FE>
0298:25 B8 D9 A6 D9 72 D9 D8<AC>
02A0:D9 55 DB 4C DB 89 D9 43<78>
02A8:DB 84 DA 93 D9 70 DB 5E<49>
02B0:DB 67 DB 79 DB CD D8 B6<5E>
02B8:DA FF DA DE D9 E2 D9 E6<68>
02C0:D9 EA D9 1A 25 69 DC 77<65>
02C8:DC B8 DD 4A DA 85 DC 2A<B3>
02D0:DA C2 D9 E4 DA C1 DA FC<4D>
02D8:FA C4 FA D8 DC A3 24 86<BF>
02E0:DA AD DA E7 DA 9E D8 AC<89>
02E8:D8 A6 D8 A3 D8 A3 24 B5<CE>
02F0:DD E4 DC E9 DC 00 00 00<45>
02F8:00 00 00 00 00 00 00 00<FC>
0300:00 00 20 20 20 20 20 20<ED>
0308:2C 28 22 22 28 1E 1A 1C<9E>
0310:32 04 32 32 30 30 30 30<6A>
0318:30 30 2E 2E 32 32 32 32<9B>
0320:32 04 32 32 32 32 32 32<58>
0328:32 32 32 32 32 32 32 32<2D>
0330:32 32 32 32 32 32 32 32<15>
0338:32 32 32 2A 2A 32 32 2A<75>
0340:32 32 28 32 32 32 28 00<BE>
0348:32 32 32 32 32 32 32 00<1A>
0350:32 32 32 00 00 00 00 00<22>
0358:00 00 00 00 00 00 00 00<5E>
0360:20 20 20 20 20 20 2C 28<64>
0368:22 22 28 1D 1A 1C 02 30<13>
0370:01 01 30 30 30 30 30 30<02>
0378:2D 2D 02 02 02 02 02 03<3F>
0380:02 02 02 02 02 02 02 02<86>
0388:02 02 02 02 02 02 02 02<8E>
0390:02 02 02 02 02 02 02 02<96>
0398:02 2A 2A 02 32 2A 02 02<C7>
03A0:28 02 32 32 28 00 32 02<2A>
03A8:02 32 32 32 32 00 02 32<4D>
03B0:32 EE 01 D3 20 56 E4 CE<88>
03B8:01 D3 C0 00 60 68 AA CE<AF>
03C0:01 D3 68 40 2C 0F D4 10<82>
03C8:03 6C 00 02 48 8A 48 A9<2F>
03D0:24 48 A9 A4 48 BA BD 05<06>
03D8:01 48 D8 48 8A 48 98 48<CF>
03E0:EE 01 D3 8D 0F D4 6C 22<03>
03E8:02 48 A9 24 48 A9 A6 48<B3>
03F0:08 EE 01 D3 6C 16 02 EE<71>
03F8:01 D3 20 E8 24 CE 01 D3<E4>
0400:60 BD 47 03 48 BD 46 03<43>
0408:48 98 A0 5C 60 EE 01 D3<53>
0410:20 00 00 CE 01 D3 60 20<39>
0418:34 C5 EE 01 D3 AD 8E 20<C7>
0420:AC 8F 20 85 0C 84 0D 6C<73>
0428:0A 00 20 34 C5 EE 01 D3<2F>
0430:4C 71 E4 20 16 E5 EE 01<5A>
0438:D3 A0 01 B1 D4 AA 88 B1<D4>
0440:D4 CE 01 D3 85 D4 86 D5<50>
0448:4C 7D D9 20 0C E5 EE 01<83>
0450:D3 A0 00 91 9B A5 D5 C8<D4>
0458:D0 0B 4C 28 E5 20 0C E5<BB>
0460:D0 F8 EE 01 D3 91 9B AD<D7>
0468:01 D3 29 FC 09 02 8D 01<F6>
0470:D3 60 20 62 25 20 DC 2B<A0>
0478:4C 3A E5 A9 23 48 48 A5<DC>
0480:B0 85 C6 20 16 E5 C6 C6<72>
0488:30 09 A5 D4 48 A5 D5 48<A3>
0490:4C 6A 25 EE 01 D3 A5 B0<ED>
0498:48 6C D4 00 4C 7B F6 A9<3F>
04A0:00 84 A4 85 A5 18 A5 90<C5>
04A8:65 A4 A8 A5 91 65 A5 CD<CB>
04B0:E6 02 90 09 D0 E6 CC E5<61>
04B8:02 90 02 D0 DF 38 A5 90<D4>
04C0:F5 00 85 A2 A5 91 F5 01<70>
04C8:85 A3 18 B5 00 85 97 85<10>
04D0:99 65 A4 85 9B B5 01 85<7D>
04D8:98 85 9A 65 A5 85 9C B5<80>
04E0:00 65 A4 95 00 B5 01 65<29>
04E8:A5 95 01 E8 E8 E0 92 90<BB>
04F0:EE 85 0F A5 90 85 0E EE<E9>
04F8:01 D3 A6 A3 18 8A 65 9A<A2>
0500:85 9A 18 8A 65 9C 85 9C<57>
0508:E8 A4 A2 F0 25 88 B1 99<A4>
0510:91 9B 98 D0 F8 F0 1B C6<15>
0518:9A C6 9C 88 B1 99 91 9B<A1>
0520:88 B1 99 91 9B 88 B1 99<A4>
0528:91 9B 88 B1 99 91 9B 98<A2>
0530:D0 E9 CA D0 E2 CE 01 D3<AD>
0538:60 A9 00 84 A4 85 A5 38<87>
0540:A5 90 F5 00 85 A2 A5 91<C4>
0548:F5 01 85 A3 38 B5 00 85<F9>
0550:99 E5 A4 85 9B B5 01 85<DB>
0558:9A E5 A5 85 9C 38 B5 00<8C>
0560:E5 A4 95 00 B5 01 E5 A5<9A>
0568:95 01 E8 E8 E0 92 90 ED<D1>
0570:85 0F A5 90 85 0E EE 01<37>
0578:D3 A0 00 A6 A3 F0 1D B1<7E>
0580:99 91 9B C8 B1 99 91 9B<33>
0588:C8 B1 99 91 9B C8 B1 99<61>
0590:91 9B C8 D0 EA E6 9A E6<08>
0598:9C CA D0 E3 A6 A2 F0 08<4B>
05A0:B1 99 91 9B C8 CA D0 F8<42>
05A8:CE 01 D3 60 18 4C 9C 2B<8C>
05B0:18 60 38 60 20 9C 2E A5<40>
05B8:D4 F0 F5 A5 E0 F0 ED 45<6F>
05C0:D4 29 80 85 EE A5 E0 29<21>
05C8:7F 85 E0 A5 D4 29 7F 38<33>
05D0:E9 40 38 65 E0 30 DB 05<13>
05D8:EE A8 20 9E 2A 85 DA 85<3D>
05E0:DB 85 DC 85 DD 85 DE 85<CB>
05E8:DF 84 D4 A0 07 46 E5 90<7A>
05F0:30 18 A5 DE 79 F2 22 85<67>
05F8:DE A5 DD 79 EA 22 85 DD<8B>
0600:A5 DC 79 E2 22 85 DC A5<88>
0608:DB 79 DA 22 85 DB A5 DA<F0>
0610:79 D2 22 85 DA A5 D9 79<7A>
0618:CA 22 85 D9 88 10 CE 30<05>
0620:05 F0 03 88 10 C7 A0 07<69>
0628:46 E4 90 30 18 A5 DD 79<CB>
0630:F2 22 85 DD A5 DC 79 EA<8D>
0638:22 85 DC A5 DB 79 E2 22<2B>
0640:85 DB A5 DA 79 DA 22 85<8C>
0648:DA A5 D9 79 D2 22 85 D9<54>
0650:A5 D8 79 CA 22 85 D8 88<CF>
0658:10 CE 30 05 F0 03 88 10<C0>
0660:C7 A0 07 46 E3 90 30 18<0E>
0668:A5 DC 79 F2 22 85 DC A5<B0>
0670:DB 79 EA 22 85 DB A5 DA<1A>
0678:79 E2 22 85 DA A5 D9 79<16>
0680:DA 22 85 D9 A5 D8 79 D2<D1>
0688:22 85 D8 A5 D7 79 CA 22<6A>
0690:85 D7 88 10 CE 30 05 F0<30>
0698:03 88 10 C7 A0 07 46 E2<FC>
06A0:90 30 18 A5 DB 79 F2 22<E0>
06A8:85 DB A5 DA 79 EA 22 85<9F>
06B0:DA A5 D9 79 E2 22 85 D9<EF>
06B8:A5 D8 79 DA 22 85 D8 A5<51>
06C0:D7 79 D2 22 85 D7 A5 D6<5B>
06C8:79 CA 22 85 D6 88 10 CE<D3>
06D0:30 05 F0 03 88 10 C7 A0<0E>
06D8:07 46 E1 90 30 18 A5 DA<1C>
06E0:79 F2 22 85 DA A5 D9 79<C2>
06E8:EA 22 85 D9 A5 D8 79 E2<F0>
06F0:22 85 D8 A5 D7 79 DA 22<A2>
06F8:85 D7 A5 D6 79 D2 22 85<91>
0700:D6 A5 D5 79 CA 22 85 D5<99>
0708:88 10 CE 30 05 F0 03 88<91>
0710:10 C7 4C B4 2D 18 60 38<4F>
0718:60 A5 E0 F0 FA A5 D4 F0<B4>
0720:F4 45 E0 29 80 85 EE A5<A4>
0728:E0 29 7F 85 E0 A5 D4 29<04>
0730:7F 38 E5 E0 18 69 40 30<79>
0738:DE 05 EE A8 20 9E 2A 85<0F>
0740:E6 85 E7 85 E8 85 E9 85<90>
0748:EA 85 E0 85 DA 84 D4 A0<1D>
0750:00 A5 E0 D9 CA 22 D0 28<B7>
0758:A5 E1 D9 D2 22 D0 21 A5<60>
0760:E2 D9 DA 22 D0 1A A5 E3<06>
0768:D9 E2 22 D0 13 A5 E4 D9<6C>
0770:EA 22 D0 0C A5 E5 D9 F2<13>
0778:22 D0 05 A2 00 4C 95 2A<CA>
0780:90 2A A5 E5 F9 F2 22 85<BF>
0788:E5 A5 E4 F9 EA 22 85 E4<45>
0790:A5 E3 F9 E2 22 85 E3 A5<F1>
0798:E2 F9 DA 22 85 E2 A5 E1<11>
07A0:F9 D2 22 85 E1 A5 E0 F9<6E>
07A8:CA 22 85 E0 26 D5 C8 C0<AA>
07B0:08 D0 9E A0 00 A5 E1 D9<D5>
07B8:CA 22 D0 28 A5 E2 D9 D2<D9>
07C0:22 D0 21 A5 E3 D9 DA 22<92>
07C8:D0 1A A5 E4 D9 E2 22 D0<5A>
07D0:13 A5 E5 D9 EA 22 D0 0C<3F>
07D8:A5 E6 D9 F2 22 D0 05 A2<E9>
07E0:01 4C 95 2A 90 2A A5 E6<EC>
07E8:F9 F2 22 85 E6 A5 E5 F9<11>
07F0:EA 22 85 E5 A5 E4 F9 E2<E4>
07F8:22 85 E4 A5 E3 F9 DA 22<21>
0800:85 E3 A5 E2 F9 D2 22 85<26>
0808:E2 A5 E1 F9 CA 22 85 E1<E0>
0810:26 D6 C8 C0 08 D0 9E A0<08>
0818:00 A5 E2 D9 CA 22 D0 28<B9>
0820:A5 E3 D9 D2 22 D0 21 A5<EB>
0828:E4 D9 DA 22 D0 1A A5 E5<F3>
0830:D9 E2 22 D0 13 A5 E6 D9<36>
0838:EA 22 D0 0C A5 E7 D9 F2<BC>
0840:22 D0 05 A2 02 4C 95 2A<A4>
0848:90 2A A5 E7 F9 F2 22 85<26>
0850:E7 A5 E6 F9 EA 22 85 E6<50>
0858:A5 E5 F9 E2 22 85 E5 A5<C3>
0860:E4 F9 DA 22 85 E4 A5 E3<FE>
0868:F9 D2 22 85 E3 A5 E2 F9<60>
0870:CA 22 85 E2 26 D7 C8 C0<19>
0878:08 D0 9E A0 00 A5 E3 D9<8F>
0880:CA 22 D0 28 A5 E4 D9 D2<37>
0888:22 D0 21 A5 E5 D9 DA 22<5C>
0890:D0 1A A5 E6 D9 F2 22 D0<64>
0898:13 A5 E7 D9 EA 22 D0 0C<49>
08A0:A5 E8 D9 F2 22 D0 05 A2<6F>
08A8:03 4C 95 2A 90 2A A5 E8<B0>
08B0:F9 F2 22 85 E8 A5 E7 F9<E0>
```

```
08B8:EA 22 85 E7 A5 E6 F9 E2<B5>
08C0:22 85 E6 A5 E5 F9 DA 22<0A>
08C8:85 E5 A5 E4 F9 D2 22 85<FF>
08D0:E4 A5 E3 F9 CA 22 85 E3<F3>
08D8:26 D8 C8 C0 08 D0 9E A0<43>
08E0:00 A5 E4 D9 CA 22 D0 28<C2>
08E8:A5 E5 D9 D2 22 D0 21 A5<14>
08F0:E6 D9 DA 22 D0 1A A5 E7<1E>
08F8:D9 E2 22 D0 13 A5 E8 D9<92>
0900:EA 22 D0 0C A5 E9 D9 F2<BE>
0908:22 D0 05 A2 04 4C 95 2A<7E>
0910:90 2A A5 E9 F9 F2 22 85<78>
0918:E9 A5 E8 F9 EA 22 85 E8<1B>
0920:A5 E7 F9 E2 22 85 E7 A5<55>
0928:E6 F9 DA 22 85 E6 A5 E5<BC>
0930:F9 D2 22 85 E5 A5 E4 F9<4A>
0938:CA 22 85 E4 26 D9 C8 C0<6A>
0940:08 D0 9E A5 D5 D0 64 A0<BF>
0948:00 A5 E5 D9 CA 22 D0 28<5C>
0950:A5 E6 D9 D2 22 D0 21 A5<9D>
0958:E7 D9 DA 22 D0 1A A5 E8<7F>
0960:D9 E2 22 D0 13 A5 E9 D9<72>
0968:EA 22 D0 0C A5 EA D9 F2<2B>
0970:22 D0 05 A2 05 4C 95 2A<EE>
0978:90 2A A5 EA F9 F2 22 85<E9>
0980:EA A5 E9 F9 EA 22 85 E9<67>
0988:A5 E8 F9 E2 22 85 E8 A5<40>
0990:E7 F9 DA 22 85 E7 A5 E6<98>
0998:F9 D2 22 85 E6 A5 E5 F9<B4>
09A0:CA 22 85 E5 26 DA C8 C0<13>
09A8:08 D0 9E 4C B4 2D 36 D5<BE>
09B0:C8 C0 08 D0 F9 F0 F4 F8<FC>
09B8:18 A5 E5 8D F9 22 65 E5<83>
09C0:8D F8 22 A5 E4 8D F1 22<FE>
09C8:65 E4 8D F0 22 A5 E3 8D<AD>
09D0:E9 22 65 E3 8D E8 22 A5<8F>
09D8:E2 8D E1 22 65 E2 8D E0<23>
09E0:22 A5 E1 8D D9 22 65 E1<CC>
09E8:8D D8 22 A9 00 8D D1 22<3C>
09F0:69 00 8D D0 22 A2 02 BD<9D>
09F8:F6 22 7D F6 22 9D F5 22<2C>
0A00:BD EE 22 7D EE 22 9D ED<AA>
0A08:22 BD E6 22 7D E6 22 9D<A0>
0A10:E5 22 BD DE 22 7D DE 22<79>
0A18:9D DD 22 BD D6 22 7D D6<48>
0A20:22 9D D5 22 BD CE 22 7D<4F>
0A28:CE 22 9D CD 22 CA D0 C7<CB>
0A30:AD F8 22 6D F6 22 8D F5<E9>
0A38:22 AD F0 22 6D EE 22 8D<9F>
0A40:ED 22 AD E8 22 6D E6 22<15>
0A48:8D E5 22 AD E0 22 6D DE<1B>
0A50:22 8D DD 22 AD D8 22 6D<CC>
0A58:D6 22 8D D5 22 AD D0 22<54>
0A60:6D CE 22 8D CD 22 A2 02<51>
0A68:BD F3 22 7D F3 22 9D F2<EE>
0A70:22 BD EB 22 7D EB 22 9D<B7>
0A78:EA 22 BD E3 22 7D E3 22<C6>
0A80:9D E2 22 BD DB 22 7D DB<0F>
0A88:22 9D DA 22 BD D3 22 7D<28>
0A90:D3 22 9D D2 22 BD CB 22<FE>
0A98:7D CB 22 9D CA 22 CA 10<96>
0AA0:C7 A5 D5 85 E1 A5 D6 85<0E>
0AA8:E2 A5 D7 85 E3 A5 D8 85<EC>
0AB0:E4 A5 D9 85 E5 A9 00 85<92>
0AB8:D4 85 D5 85 D6 85 D7 85<C2>
0AC0:D8 85 D9 60 A9 05 85 F4<AB>
0AC8:A9 80 85 F3 60 EE 01 D3<60>
0AD0:20 A1 DB CE 01 D3 60 EE<57>
0AD8:01 D3 20 E6 D8 CE 01 D3<4F>
0AE0:60 EE 01 D3 20 00 D8 CE<FC>
0AE8:01 D3 60 A4 F2 B1 F3 38<EC>
0AF0:E9 30 C9 0A 60 A4 D4 A5<76>
0AF8:D5 85 F7 20 9C 2B F8 98<12>
0B00:F0 2E 4A 4A 4A 4A 85 F8<DD>
0B08:98 29 07 A0 00 90 02 69<4A>
0B10:07 46 F8 90 02 69 15 46<55>
0B18:F8 90 02 69 31 46 F8 90<F7>
0B20:05 69 63 90 01 C8 46 F8<E7>
0B28:90 06 69 27 C8 90 01 C8<71>
0B30:A6 F7 F0 6F 46 F7 90 07<FF>
0B38:69 55 C8 C8 90 01 C8 46<B3>
0B40:F7 90 08 69 11 AA 98 69<56>
0B48:05 A8 8A 46 F7 90 08 69<E3>
0B50:23 AA 98 69 10 A8 8A 46<88>
0B58:F7 90 08 69 47 AA 98 69<D0>
0B60:20 A8 8A 46 F7 90 08 69<38>
0B68:95 AA 98 69 40 A8 8A 46<F6>
0B70:F7 90 0C 69 91 AA 98 69<01>
0B78:81 A8 8A 90 02 E6 D5 46<3A>
0B80:F7 90 0E 69 83 AA 98 69<E1>
0B88:63 A8 A5 D5 69 01 85 D5<89>
0B90:8A 46 F7 90 0E 69 67 AA<E2>
0B98:98 69 27 A8 A5 D5 69 03<C2>
0BA0:85 D5 8A 84 D6 85 D7 A9<57>
0BA8:42 85 D4 4C B8 2D 18 60<5A>
0BB0:A5 E0 49 80 85 E0 A5 E0<07>
0BB8:29 7F F0 F2 85 F7 A5 D4<47>
0BC0:29 7F 38 E5 F7 B0 33 A5<E6>
0BC8:D4 A4 E0 85 E0 84 D4 A5<93>
0BD0:D5 A4 E1 85 E1 84 D5 A5<54>
0BD8:D6 A4 E2 85 E2 84 D6 A5<D5>
0BE0:D7 A4 E3 85 E3 84 D7 A5<77>
0BE8:D8 A4 E4 85 E4 84 D8 A5<32>
0BF0:D9 A4 E5 85 E5 84 D9 4C<FB>
0BF8:9D 2C A8 F0 4B 88 F0 36<8A>
0C00:88 F0 22 88 F0 10 88 D0<B5>
0C08:A5 A5 E1 85 E5 84 E4 84<09>
0C10:E3 84 E2 4C 2D 2D A5 E2<05>
0C18:85 E5 A5 E1 85 E4 84 E3<7E>
0C20:84 E2 4C 2D 2D A5 E3 85<92>
0C28:E5 A5 E2 85 E4 A5 E1 85<62>
0C30:E3 84 E2 4C 2D 2D A5 E4<E7>
0C38:85 E5 A5 E3 85 E4 A5 E2<7F>
0C40:85 E3 A5 E1 85 E2 84 E1<89>
0C48:F8 A5 D4 45 E0 30 3A 18<9C>
0C50:A5 D9 65 E5 85 D9 A5 D8<98>
0C58:65 E4 85 D8 A5 D7 65 E3<DF>
0C60:85 D7 A5 D6 65 E2 85 D6<F4>
0C68:A5 D5 65 E1 85 D5 90 16<F9>
0C70:A5 D8 85 D9 A5 D7 85 D8<E3>
0C78:A5 D6 85 D7 A5 D5 85 D6<2E>
0C80:A9 01 85 D5 E6 D4 4C B8<7A>
0C88:2D 38 A5 D9 E5 E5 85 D9<E6>
0C90:A5 D8 E5 E4 85 D8 A5 D7<F4>
0C98:E5 E3 85 D7 A5 D6 E5 E2<0A>
0CA0:85 D6 A5 D5 E5 E1 85 D5<5B>
0CA8:B0 27 A5 D4 49 80 85 D4<7B>
0CB0:38 98 E5 D9 85 D9 98 E5<92>
0CB8:D8 85 D8 98 E5 D7 85 D7<B6>
0CC0:98 E5 D6 85 D6 98 E5 D5<94>
0CC8:85 D5 4C B8 2D A6 DA D0<D1>
0CD0:02 A2 00 D8 A4 D4 F0 6D<99>
0CD8:A5 D5 D0 5C 88 A5 D6 D0<CE>
0CE0:47 88 A5 D7 D0 30 88 A5<3B>
0CE8:D8 D0 1B 88 A5 D9 D0 08<7C>
0CF0:88 8A F0 51 85 D5 D0 40<51>
0CF8:85 D5 86 D6 A9 00 85 D9<5D>
0D00:85 D8 85 D7 F0 32 85 D5<F7>
0D08:A5 D9 85 D6 86 D7 A9 00<1A>
0D10:85 D9 85 D8 F0 22 85 D5<37>
0D18:A5 D8 85 D6 A5 D9 85 D7<19>
0D20:86 D8 A9 00 85 D9 F0 10<09>
0D28:85 D5 A5 D7 85 D6 A5 D8<9C>
0D30:85 D7 A5 D9 85 D8 86 D9<EE>
0D38:84 D4 98 29 7F C9 71 B0<56>
0D40:08 C9 0F B0 03 20 9C 2B<7A>
0D48:18 60 86 FC 84 FD 85 EF<E0>
0D50:20 B5 2E 20 7C 2E C6 EF<7C>
0D58:20 9E 26 B0 37 A5 FC 69<16>
0D60:06 85 FC 90 02 E6 FD 20<D8>
0D68:7C 2E 20 9D 2C B0 25 C6<5A>
0D70:EF F0 21 AD E0 05 85 E0<1B>
0D78:AD E1 05 85 E1 AD E2 05<1D>
0D80:85 E2 AD E3 05 85 E3 AD<4A>
0D88:E4 05 85 E4 AD E5 05 85<2C>
0D90:E5 4C 3F 2E 60 A0 05 B1<37>
0D98:FC 85 E5 88 B1 FC 85 E4<4B>
0DA0:88 B1 FC 85 E3 88 B1 FC<87>
0DA8:85 E2 88 B1 FC 85 E1 88<F9>
0DB0:B1 FC 85 E0 60 A5 D4 85<93>
0DB8:E0 A5 D5 85 E1 A5 D6 85<E9>
0DC0:E2 A5 D7 85 E3 A5 D8 85<50>
0DC8:E4 A5 D9 85 E5 60 A5 D4<69>
0DD0:8D E0 05 A5 D5 8D E1 05<B3>
0DD8:A5 D6 8D E2 05 A5 D7 8D<A0>
0DE0:E3 05 A5 D8 8D E4 05 A5<76>
0DE8:D9 8D E5 05 60 A5 D4 8D<C1>
0DF0:E6 05 A5 D5 8D E7 05 A5<57>
0DF8:D6 8D E8 05 A5 D7 8D E9<92>
0E00:05 A5 D8 8D EA 05 A5 D9<21>
0E08:8D EB 05 60 A2 00 4C FF<F0>
0E10:2E A2 06 4C FF 2E A2 0C<63>
0E18:BD E0 05 85 D4 BD E1 05<CB>
0E20:85 D5 BD E2 05 85 D6 BD<8E>
0E28:E3 05 85 D7 BD E4 05 85<F9>
0E30:D8 BD E5 05 85 D9 60 A2<40>
0E38:0C 4C 25 2F A2 06 BD E0<85>
0E40:05 85 E0 BD E1 05 85 E1<ED>
0E48:BD E2 05 85 E2 BD E3 05<31>
0E50:85 E3 BD E4 05 85 E4 BD<20>
0E58:E5 05 85 E5 60 A2 05 EE<A0>
0E60:01 D3 BD 89 DE 95 E0 CA<A0>
0E68:10 F8 CE 01 D3 20 9E 26<CC>
0E70:B0 EA A9 00 85 F1 A5 D4<7B>
0E78:85 F0 29 7F 85 D4 C9 40<B1>
0E80:90 1E D0 D8 A5 D5 29 F0<9E>
0E88:4A 85 F1 4A 4A 65 F1 85<34>
0E90:F1 A5 D5 29 0F 65 F1 85<EC>
0E98:F1 A9 00 85 D5 20 B8 2D<73>
0EA0:A9 0A A2 4D A0 DE EE 01<ED>
0EA8:D3 20 31 2E CE 01 D3 20<77>
0EB0:9B 26 A5 F1 F0 0F 4A 18<CF>
0EB8:65 D4 30 16 85 D4 46 F1<7C>
0EC0:90 03 20 9B 32 06 F0 90<2C>
0EC8:0A 20 9C 2E 20 BB 2F 4C<DE>
0ED0:00 28 38 60 A9 40 85 D4<19>
0ED8:A0 01 84 D5 88 84 D6 84<64>
0EE0:D7 84 D8 84 D9 60 86 FC<D9>
0EE8:84 FD 20 B5 2E 20 7C 2E<ED>
0EF0:20 9D 2C 20 D4 2E 20 F3<55>
0EF8:2E 20 7C 2E 20 97 2C 20<59>
0F00:23 2F 4C 00 28 38 60 A9<70>
0F08:05 D0 02 A9 00 85 F0 A5<FE>
0F10:D4 30 F2 F0 F0 0A 49 80<1B>
0F18:85 F1 A9 40 85 D4 A5 D5<E0>
0F20:29 F0 F0 05 E6 F1 20 A3<F3>
0F28:32 A2 66 A0 DF EE 01 D3<22>
0F30:20 CD 2F 20 D4 2E 20 9B<AC>
0F38:26 A9 0A A2 72 A0 DF 20<E6>
0F40:31 2E CE 01 D3 20 23 2F<B8>
0F48:20 9E 26 A9 3F 85 E0 A9<91>
0F50:50 85 E1 A9 00 85 E2 85<89>
0F58:E3 85 E4 85 E5 20 9D 2C<EA>
0F60:20 9C 2E A5 F1 10 05 18<2F>
0F68:49 FF 69 01 85 D4 A9 00<F8>
0F70:85 D5 20 DC 2B A5 F1 29<3F>
0F78:80 05 D4 85 D4 20 9D 2C<71>
0F80:A6 F0 F0 11 EE 01 D3 BD<D3>
0F88:89 DE 95 E0 CA 10 F8 CE<E0>
0F90:01 D3 4C 00 28 18 60 38<07>
0F98:60 A9 04 24 D4 10 06 A9<67>
0FA0:02 D0 02 A9 01 85 F0 A5<2D>
0FA8:D4 29 7F 85 D4 A6 FB BD<FB>
0FB0:36 31 85 E0 BD 37 31 85<6E>
0FB8:E1 BD 38 31 85 E2 BD 39<28>
0FC0:31 85 E3 BD 3A 31 85 E4<44>
0FC8:BD 3B 31 85 E5 20 00 28<52>
0FD0:B0 C5 A5 D4 29 7F 38 E9<A6>
0FD8:40 30 24 C9 04 10 B8 AA<2B>
0FE0:B5 D5 85 F1 29 10 F0 02<B9>
0FE8:A9 02 18 65 F1 29 03 65<3B>
0FF0:F0 85 F0 86 F1 A9 00 95<90>
0FF8:D5 CA 10 FB 20 B8 2D 46<65>
1000:F0 90 09 20 9C 2E 20 BB<F0>
1008:2F 20 97 2C 20 D4 2E 20<09>
1010:9B 26 B0 83 A9 06 A2 18<3E>
1018:A0 31 20 31 2E 20 23 2F<D2>
1020:20 9E 26 46 F0 90 09 18<FC>
1028:A5 D4 F0 04 49 80 85 D4<F1>
1030:60 BD 03 55 14 99 39 3E<C8>
1038:01 60 44 27 52 BE 46 81<81>
1040:75 43 55 3F 07 96 92 62<1F>
1048:39 BF 64 59 64 08 67 40<BE>
1050:01 57 07 96 32 40 90 00<72>
1058:00 00 00 3F 01 74 53 29<15>
1060:25 A9 00 85 F0 85 F1 A5<E2>
1068:D4 29 7F C9 40 30 1B A5<33>
1070:D4 29 80 85 F0 E6 F1 A9<0F>
1078:7F 25 D4 85 D4 A2 EA A0<1D>
1080:DF EE 01 D3 20 CD 2F CE<39>
1088:01 D3 20 D4 2E 20 9B 26<54>
1090:B0 47 A9 0B A2 AE A0 DF<FE>
1098:EE 01 D3 20 31 2E CE 01<2B>
10A0:D3 B0 36 20 23 2F 20 9E<8E>
10A8:26 B0 2E A5 F1 F0 19 A2<F3>
10B0:05 EE 01 D3 BD F0 DF 95<14>
10B8:E0 CA 10 F8 CE 01 D3 20<35>
10C0:9D 2C A5 F0 05 D4 85 D4<22>
10C8:A5 FB F0 0D A2 05 BD 42<AD>
10D0:31 95 E0 CA 10 F8 20 00<A7>
10D8:28 60 38 60 18 60 A9 00<35>
10E0:85 F1 A5 D4 30 F4 F0 F4<36>
10E8:C9 3F F0 05 18 69 01 85<F4>
10F0:F1 A9 06 85 EF A9 3F 85<29>
10F8:D4 20 D4 2E 20 9C 2E 20<4F>
1100:BB 2F E6 D5 20 97 2C 20<69>
1108:23 2F 20 9E 26 A5 D4 8D<1D>
1110:EC 05 A5 D5 8D ED 05 A5<3A>
1118:D6 8D EE 05 A5 D7 8D EF<E0>
```

Listing »Turbo-Basic XL«.
Bitte unbedingt mit dem Eingabe-Programm AMPEL von Seite 63 eintippen

```
1120:05 A5 D8 8D F0 05 A5 D9<A7>
1128:8D F1 05 20 9C 2E 20 F8<3D>
1130:2E 20 00 28 20 1E 2F 20<EB>
1138:97 2C 20 4F 32 A5 D4 F0<65>
1140:0C 20 1E 2F 20 9D 2C C6<B4>
1148:EF 10 C2 30 03 20 FD 2E<F1>
1150:A5 F1 F0 0F 4A 18 65 D4<51>
1158:E9 1F 85 D4 46 F1 90 03<9E>
1160:20 9B 32 18 60 4C 9C 2B<C7>
1168:A5 D4 29 7F C9 0F 90 F5<06>
1170:20 9C 2E F8 A2 00 A0 04<1F>
1178:18 A5 D9 65 E5 85 D9 A5<C5>
1180:D8 65 E4 85 D8 A5 D7 65<22>
1188:E3 85 D7 A5 D6 65 E2 85<2F>
1190:D6 A5 D5 65 E1 85 D5 8A<04>
1198:69 00 AA 88 D0 DB D8 8A<50>
11A0:F0 1A A5 D8 85 D9 A5 D7<E3>
11A8:85 D8 A5 D6 85 D7 A5 D5<B4>
11B0:85 D6 86 D5 A5 D5 C9 10<CA>
11B8:90 31 E6 D4 A5 D5 C9 10<0A>
11C0:90 27 4A 66 D6 66 D7 66<6C>
11C8:D8 66 D9 4A 66 D6 66 D7<4E>
11D0:66 D8 66 D9 4A 66 D6 66<CB>
11D8:D7 66 D8 66 D9 4A 66 D6<E2>
11E0:66 D7 66 D8 66 D9 85 D5<A5>
11E8:60 C6 D4 A9 00 06 D9 26<A0>
11F0:D8 26 D7 26 D6 26 D5 2A<1D>
11F8:06 D9 26 D8 26 D7 26 D6<44>
1200:26 D5 2A 06 D9 26 D8 26<47>
1208:D7 26 D6 26 D5 2A 06 D9<92>
1210:26 D8 26 D7 26 D6 26 D5<41>
1218:2A 60 C9 FF 90 02 C0 50<CE>
1220:8A 69 00 85 D4 60 98 29<34>
1228:F0 4A 85 D4 4A 4A 65 D4<BA>
1230:85 D4 98 29 0F A6 D6 E0<BD>
1238:50 65 D4 85 D4 60 A2 00<89>
1240:A4 D5 A5 D4 86 D5 38 E9<26>
1248:40 90 CF F0 D9 C9 02 F0<EE>
1250:45 B0 42 A5 D6 29 F0 4A<8E>
1258:85 D4 4A 4A 65 D4 85 D4<EB>
1260:A5 D6 29 0F A6 D7 E0 50<7F>
1268:65 D4 85 D4 98 29 0F AA<33>
1270:BD CD 33 65 D4 85 D4 BD<6A>
1278:D7 33 65 D5 85 D5 98 29<90>
1280:F0 F0 12 4A 4A 4A AA A5<EB>
1288:D4 7D B9 33 85 D4 A5 D5<CD>
1290:7D BA 33 85 D5 60 C0 07<43>
1298:B0 FB A5 D7 29 F0 4A 85<77>
12A0:D4 4A 4A 65 D4 85 D4 A5<95>
12A8:D7 29 0F A6 D8 E0 50 65<D2>
12B0:D4 85 D4 B9 AD 33 65 D4<B2>
12B8:85 D4 B9 B4 33 65 D5 85<C0>
12C0:D5 A4 D6 D0 A7 60 00 10<FE>
12C8:20 30 40 50 60 00 27 4E<6D>
12D0:75 9C C3 EA E8 03 D0 07<00>
12D8:B8 0B A0 0F 88 13 70 17<F8>
12E0:58 1B 40 1F 28 23 00 64<12>
12E8:C8 2C 90 F4 58 BC 20 84<22>
12F0:00 00 00 01 01 01 02 02<36>
12F8:03 03 4C 28 E5 20 1E F4<20>
1300:20 05 E5 85 A2 84 A3 05<D5>
1308:A3 F0 EF 98 05 9A 05 9C<87>
1310:30 E8 A5 99 85 A0 A5 9A<0C>
1318:85 A1 20 62 C9 A5 8A 85<77>
1320:E2 A5 8B 85 E3 A9 80 85<6B>
1328:D5 0A 85 D4 20 0E 35 30<DC>
1330:C9 A5 89 85 8B A5 88 85<EA>
1338:8A A0 01 B1 8A 30 44 C8<93>
1340:B1 8A 85 9F C8 B1 8A 85<66>
1348:A7 C8 84 A8 B1 8A C9 0A<3A>
1350:F0 0A C9 0B F0 06 C9 0C<F1>
1358:F0 02 C9 0D F0 69 C9 1E<B7>
1360:F0 6B C9 23 F0 5A C9 07<66>
1368:F0 4D C9 04 F0 6B C9 56<45>
1370:F0 67 A4 A7 C4 9F 90 CD<EE>
1378:18 A5 8A 65 9F 90 B8 E6<E8>
1380:8B B0 B4 A5 E3 85 8B A5<02>
1388:E2 85 8A A0 01 B1 8A 30<2F>
1390:20 A5 9C 91 8A 88 A5 9B<0F>
1398:91 8A 18 65 A2 85 9B A5<F2>
13A0:9C 65 A3 85 9C A0 02 B1<B1>
13A8:8A 65 8A 90 DC E6 8B B0<72>
13B0:D8 20 C5 C9 4C B5 F1 20<2F>
13B8:66 35 E0 1B D0 B4 C6 A8<0D>
13C0:A4 A8 C8 C4 A7 B0 AB 20<65>
13C8:D5 34 4C 59 34 20 66 35<97>
13D0:E0 17 F0 07 E0 18 D0 9A<B4>
13D8:2C E6 A8 A5 A8 C5 A7 B0<AA>
13E0:91 48 20 D7 34 68 85 A8<42>
13E8:20 68 35 4C C2 34 E6 A8<E6>
13F0:A4 A8 84 DC B1 8A F0 2E<42>
13F8:C9 0F B0 2A 20 4F E4 20<21>
1400:25 33 A5 D5 30 20 B0 1E<C8>
1408:20 0E 35 08 B0 04 85 D4<57>
```

```
1410:84 D5 20 DC 2B 06 D4 28<B2>
1418:66 D4 A4 A8 A2 05 B5 D4<D6>
1420:88 91 8A CA 10 F8 60 A5<14>
1428:D4 C5 99 A5 D5 E5 9A B0<0E>
1430:05 A5 D4 A4 D5 60 A5 E2<30>
1438:85 DA A5 E3 85 DB A5 9B<53>
1440:85 E0 A5 9C 85 E1 A0 01<01>
1448:B1 DA 30 2D C5 D5 D0 05<5E>
1450:88 B1 DA C5 D4 B0 1F A0<3B>
1458:02 B1 DA 65 DA 85 DA 90<A4>
1460:02 E6 DB 88 B1 DA 30 11<57>
1468:18 A5 E0 65 A2 85 E0 A5<A8>
1470:E1 65 A3 4C 2B 35 18 F0<E9>
1478:01 38 A5 E0 A4 E1 60 E6<11>
1480:A8 20 4F E4 90 FB AA BD<65>
1488:CC 23 D0 F5 60 A9 00 2C<7A>
1490:A9 80 8D 16 23 A9 F6 A0<7B>
1498:35 8D E0 02 8C E1 02 A9<6F>
14A0:04 A0 01 20 C8 C2 EE 01<37>
14A8:D3 20 07 36 C9 FF D0 60<88>
14B0:C8 D0 5D A9 F6 A0 35 8D<0C>
14B8:E2 02 8C E3 02 20 07 36<C6>
14C0:C9 FF D0 04 C0 FF F0 EB<0C>
14C8:9D 44 03 98 9D 45 03 20<20>
14D0:07 36 38 FD 44 03 9D 48<50>
14D8:03 98 FD 45 03 9D 49 03<BB>
14E0:FE 48 03 D0 03 FE 49 03<1C>
14E8:20 56 E4 98 30 24 20 01<0D>
14F0:36 AD 53 03 C9 03 D0 BB<8D>
14F8:A2 10 A9 0C 9D 42 03 20<5F>
1500:56 E4 2C 16 23 10 03 20<B4>
1508:04 36 A9 FE 8D 01 D3 60<04>
1510:A9 01 A8 20 24 23 98 4C<EE>
1518:E4 DC 6C E2 02 6C E0 02<56>
1520:A9 07 A2 10 9D 42 03 A9<52>
1528:00 9D 48 03 9D 49 03 20<06>
1530:56 E4 30 DF 48 A9 00 9D<97>
1538:48 03 20 56 E4 30 D4 A8<55>
1540:68 60 00 60 7B 61 AE 24<14>
1548:E4 AC 25 E4 E8 D0 01 C8<7A>
1550:8E F8 24 8C F9 24 A9 00<56>
1558:8D 0E D4 78 A9 FE 8D 01<28>
1560:D3 A9 AB 8D FA FF A9 24<0F>
1568:8D FB FF A9 D0 8D FE FF<32>
1570:A9 24 8D FF FF A9 CC 20<4C>
1578:DB 60 A9 E0 20 DB 60 A9<33>
1580:40 8D 0E D4 58 A9 FF 8D<1D>
1588:01 D3 20 01 61 A9 00 85<2E>
1590:09 A5 0C A4 0D 8D 8E 20<FF>
1598:8C 8F 20 A9 80 A0 20 85<52>
15A0:0C 84 0D A9 FE 8D 01 D3<25>
15A8:85 CA A2 01 8E F8 03 86<C4>
15B0:09 CA 8E 44 02 20 04 E6<42>
15B8:A9 00 85 A0 85 A1 20 5F<0F>
15C0:C9 A0 1F A2 8A 20 86 25<74>
15C8:A0 1E B9 A7 60 91 97 88<D6>
15D0:10 F8 20 C5 C9 A9 FF 48<07>
15D8:A9 AE 48 20 C6 60 CE 01<95>
15E0:D3 A9 00 A8 99 00 B0 88<4C>
15E8:10 FA 4C 3A F5 00 00 19<66>
15F0:07 0D 67 14 19 25 0F 0D<93>
15F8:44 3A 41 55 54 4F 52 55<15>
1600:4E 2E 42 41 53 16 01 00<10>
1608:06 06 16 16 A9 FF 8D 01<3A>
1610:D3 A9 C0 85 6A 4A 85 0F<A2>
1618:AD 01 E4 48 AD 00 E4 48<51>
1620:60 85 D5 A0 00 84 D4 A2<8B>
1628:04 A9 FF 8D 01 D3 B1 D4<C3>
1630:99 00 21 C8 D0 F8 CE 01<4C>
1638:D3 B9 00 21 91 D4 C8 D0<83>
1640:F8 E6 D5 CA D0 E3 60 A2<AC>
1648:10 A9 D6 9D 44 03 A9 00<BA>
1650:9D 45 03 9D 49 03 A9 04<EB>
1658:9D 48 03 A9 07 9D 42 03<86>
1660:20 56 E4 30 59 A9 7C 85<8F>
1668:DA 9D 44 03 A9 61 85 DB<56>
1670:9D 45 03 A5 D8 E5 D6 85<EA>
1678:DC A5 D9 E5 D7 85 DD E6<D3>
1680:DC D0 02 E6 DD A5 DC 9D<FF>
1688:48 03 A5 DD 9D 49 03 20<2C>
1690:56 E4 30 2B CE 01 D3 A0<53>
1698:00 A6 DD F0 0E B1 DA 91<82>
16A0:D6 C8 D0 F9 E6 D7 E6 DB<64>
16A8:CA D0 F2 A6 DC F0 08 B1<A5>
16B0:DA 91 D6 C8 CA D0 F8 A9<8B>
16B8:FF 8D 01 D3 30 89 60 6C<A5>
16C0:0A 00 E2 02 E3 02 00 60<CD>
16C8:00 C0 F7 CB 20 1E F4 20<F1>
16D0:CB C0 20 FB E4 85 A0 84<FE>
16D8:A1 20 FB E4 20 5F C9 A0<26>
16E0:01 B1 8A C5 D5 D0 05 88<EE>
16E8:B1 8A C5 D4 90 02 D0 16<5F>
16F0:A0 02 18 B1 8A A8 65 8A<40>
16F8:85 8A 90 02 E6 8B A2 8A<A2>
```

```
1700:20 20 26 4C 13 C0 20 26<89>
1708:C1 4C B5 F1 C8 C4 A7 B0<AB>
1710:03 20 BE C2 A9 00 85 B9<84>
1718:20 94 C2 A5 11 F0 34 A5<63>
1720:B9 20 B3 E4 A5 9D C5 88<F4>
1728:A5 9E E5 89 B0 27 A5 B9<51>
1730:20 6B C0 E6 B9 D0 E1 20<56>
1738:B1 F3 20 02 F2 A5 D2 C9<2E>
1740:C0 B0 1E C9 80 B0 38 C9<C6>
1748:40 B0 34 A9 3D 20 96 C2<6A>
1750:4C 59 F9 C6 11 A5 B5 F0<6A>
1758:07 20 F8 C4 A9 00 85 B5<62>
1760:60 A0 4F 4A B0 0A A0 5D<77>
1768:4A B0 05 A9 3F 4C 96 C2<EB>
1770:98 20 4B F3 A0 00 B1 D4<BF>
1778:AA C8 B1 D4 4C 52 F9 A5<7F>
1780:D9 48 A5 D8 48 A5 D7 A6<71>
1788:D6 20 52 F9 A9 2C 20 96<4F>
1790:C2 68 AA 68 4C 52 F9 8A<88>
1798:48 A5 90 48 A5 91 48 A5<2D>
17A0:B2 48 A5 A0 48 A5 A1 48<A6>
17A8:20 D3 F7 B0 2F A4 A1 D0<8E>
17B0:08 A9 80 85 A0 A9 C1 85<0B>
17B8:A1 A8 30 0E A0 00 B1 A0<0E>
17C0:C8 91 90 B1 A0 C8 91 90<03>
17C8:90 DE A0 00 B1 A0 A0 13<8A>
17D0:91 90 A0 01 B1 A0 A0 14<EA>
17D8:91 90 90 CC 68 85 A1 68<A4>
17E0:85 A0 68 85 B2 68 85 91<03>
17E8:85 0F 68 85 90 85 0E 68<EC>
17F0:AA 60 48 A5 90 48 A5 91<24>
17F8:48 A5 B2 48 A5 A0 48 A5<47>
1800:A1 48 A5 8A 48 A5 8B 48<FB>
1808:A5 BE 48 A5 BF 48 20 82<52>
1810:C1 20 C5 C9 20 D3 F7 B0<C8>
1818:24 48 A4 A1 C8 F0 05 20<23>
1820:62 C9 90 06 A9 00 85 8A<9E>
1828:85 8B A0 01 68 10 02 A0<A9>
1830:13 A5 8A 91 90 C8 A5 8B<CD>
1838:91 90 4C 48 C1 68 85 BF<04>
1840:68 85 BE 68 85 8B 68 85<3D>
1848:8A 4C 10 C1 FF FF A5 86<F3>
1850:85 A0 A5 87 85 A1 A5 A0<14>
1858:C5 88 A5 A1 E5 89 B0 19<BD>
1860:A0 00 B1 A0 29 C0 C9 C0<AA>
1868:D0 02 91 A0 18 A5 A0 69<3A>
1870:08 85 A0 90 E1 E6 A1 B0<62>
1878:DD 60 20 05 E5 29 0F 06<73>
1880:9B 06 9B 06 9B 06 9B 05<DC>
1888:9B A6 9A D0 15 A6 99 E0<3C>
1890:05 B0 0F 9D C4 02 60 A2<B2>
1898:07 A9 00 9D 00 D2 CA 10<6F>
18A0:FA 60 4C 28 E5 85 A2 C8<2A>
18A8:C4 A7 B0 EB 20 23 E5 A0<20>
18B0:00 24 A2 10 03 0A A0 78<36>
18B8:C9 04 B0 E6 0A 48 8C 08<31>
18C0:D2 A9 03 8D 0F D2 20 05<42>
18C8:E5 68 AA A5 99 9D 00 D2<EE>
18D0:24 A2 10 07 E8 E8 A5 9A<F1>
18D8:9D 00 D2 A5 9B 0A 0A 0A<86>
18E0:0A 05 D4 9D 01 D2 60 20<D8>
18E8:13 E5 85 55 84 56 20 23<91>
18F0:E5 85 54 60 20 13 E5 85<10>
18F8:C8 60 20 13 E5 8D FD 02<B2>
1900:60 A9 12 2C A9 11 48 20<C3>
1908:1B C2 A5 C8 8D FB 02 A2<87>
1910:60 A9 0C 9D 4A 03 A9 00<91>
1918:9D 4B 03 68 4C FE C5 A2<C8>
1920:06 86 C1 20 F8 C4 20 13<25>
1928:E5 A2 0D A0 23 86 F3 84<AA>
1930:F4 A2 06 29 F0 49 1C A8<A0>
1938:A5 D4 20 B4 C3 4C B9 C4<73>
1940:20 1B C2 A4 C8 A2 60 4C<1C>
1948:9C C2 A6 B4 D0 05 A5 C2<8E>
1950:20 96 C2 A6 B4 A9 05 20<49>
1958:AD C2 20 02 C5 4C B9 C4<82>
1960:A9 9B A6 B5 A8 20 AF C2<FB>
1968:BD 4A 03 85 2A BD 4B 03<52>
1970:85 2B 20 DE 24 98 4C BF<E7>
1978:C4 85 C0 86 C1 4C AB C4<9E>
1980:A9 04 20 C6 C2 85 B4 4C<45>
1988:7E E6 A9 08 20 C6 C2 85<29>
1990:B5 60 A0 07 84 C1 48 20<71>
1998:AB C4 20 FD C4 A0 03 84<0B>
19A0:C0 68 A0 00 20 DC C3 A9<1F>
19A8:07 60 A9 FF 2C A9 00 48<CB>
19B0:A9 04 20 C6 C2 68 48 A9<85>
19B8:07 85 C0 85 CA 20 AB C4<C8>
19C0:A0 0E 20 04 C5 20 B9 C4<66>
19C8:AD 80 05 0D 81 05 D0 38<DA>
19D0:A2 8C 18 A5 80 7D 00 05<BE>
19D8:A8 A5 81 7D 01 05 CD E6<2B>
19E0:02 90 0A D0 05 CC E5 02<98>
19E8:90 03 4C D4 F8 95 01 94<94>
```

```
19F0:00 CA CA E0 82 B0 DB 20<74>
19F8:6E C3 20 5B F5 A9 00 85<9B>
1A00:CA 68 F0 01 60 4C 6B E6<30>
1A08:A9 00 85 CA 4C D1 F8 A9<08>
1A10:08 20 C6 C2 A9 0B 85 C0<61>
1A18:A2 80 38 B5 00 E5 80 9D<AF>
1A20:00 05 E8 B5 00 E5 81 9D<1F>
1A28:00 05 E8 E0 8E 90 EB 20<45>
1A30:AB C4 A0 0E 20 04 C5 20<52>
1A38:B9 C4 20 AB C4 A5 82 85<E9>
1A40:F3 A5 83 85 F4 AC 8D 05<2E>
1A48:88 98 AC 8C 05 20 06 C5<2E>
1A50:20 B9 C4 4C F8 C4 A9 08<40>
1A58:20 9C C3 4C 48 C3 A9 04<53>
1A60:20 9C C3 A9 00 4C EA C2<45>
1A68:48 A2 10 86 F3 A2 23 86<AC>
1A70:F4 A2 07 68 A8 A9 80 20<6F>
1A78:B4 C3 20 B9 C4 A9 07 60<39>
1A80:48 A9 03 20 AD C2 68 9D<3D>
1A88:4B 03 98 9D 4A 03 20 0D<AB>
1A90:C5 4C AB 2B 20 13 E5 2C<75>
1A98:A9 03 85 C0 20 A6 C4 20<DF>
1AA0:13 E5 48 20 13 E5 A8 68<67>
1AA8:48 98 48 A4 A8 C8 C4 A7<7E>
1AB0:B0 1A 20 00 E4 20 73 C5<52>
1AB8:20 AB C4 68 9D 4B 03 68<22>
1AC0:9D 4A 03 20 02 C5 20 93<32>
1AC8:C5 4C B9 C4 A9 07 A2 23<8D>
1AD0:A0 05 20 71 DA 4C E9 C3<39>
1AD8:20 A6 C4 A9 0D 20 1A C5<8A>
1AE0:98 4C 20 C5 A9 26 20 51<32>
1AE8:C5 BD 4C 03 BC 4D 03 20<21>
1AF0:22 C5 20 AB C4 BD 4E 03<C5>
1AF8:4C 20 C5 20 A6 C4 20 13<84>
1B00:E5 20 AB C4 A5 D4 9D 4C<45>
1B08:03 A5 D5 9D 4D 03 20 13<25>
1B10:E5 20 AB C4 A5 D4 9D 4E<37>
1B18:03 A9 25 85 C0 4C 56 C5<C8>
1B20:20 9C C4 20 13 E5 A6 C1<FC>
1B28:20 98 C2 A4 A8 C8 C4 A7<55>
1B30:90 F1 60 B1 8A C9 1C F0<21>
1B38:0E 20 0D C6 20 20 C5 A4<2A>
1B40:A8 C8 C4 A7 90 F3 60 20<96>
1B48:A6 C4 20 AB C4 20 04 C6<65>
1B50:AA 98 20 BF C4 8A 20 20<5C>
1B58:C5 A4 A8 C8 C4 A7 90 EA<8F>
1B60:60 20 1B C2 A2 60 D0 E5<6D>
1B68:B1 8A C9 1C F0 04 A9 00<97>
1B70:F0 03 20 78 DD 85 C1 A5<EE>
1B78:C1 0A 0A 0A 0A AA 10 E0<6A>
1B80:A9 14 4C DC F8 20 AB C4<B0>
1B88:BD 43 03 10 D3 A0 00 8C<F0>
1B90:FE 02 C9 80 D0 09 84 11<34>
1B98:A6 CA F0 C4 4C 5D E6 A4<9F>
1BA0:C1 C9 88 F0 0F 85 B9 C0<37>
1BA8:07 D0 03 20 F8 C4 20 D1<9C>
1BB0:F5 4C DE F8 C0 07 D0 ED<66>
1BB8:A2 5D E4 C2 D0 E7 20 F8<D1>
1BC0:C4 4C 6E E6 20 AB C4 F0<3F>
1BC8:97 A9 0C 4C 1A C5 A0 FF<1D>
1BD0:A9 00 9D 49 03 98 9D 48<EA>
1BD8:03 A5 F4 A4 F3 9D 45 03<4B>
1BE0:98 9D 44 03 A5 C0 9D 42<EB>
1BE8:03 4C 98 24 A0 00 48 98<0D>
1BF0:48 20 00 E4 68 85 D5 68<F7>
1BF8:85 D4 20 DC 2B 4C C0 E5<90>
1C00:A9 00 A2 07 9D 00 D2 CA<8D>
1C08:10 FA A0 07 84 C1 20 F8<AA>
1C10:C4 C6 C1 D0 F9 60 C8 C4<4D>
1C18:A7 B0 E5 A9 0C 85 C0 20<C8>
1C20:A6 C4 20 18 C5 4C B9 C4<12>
1C28:A2 06 86 F2 BD 6C C5 20<C5>
1C30:96 C2 A6 F2 CA 10 F3 60<6C>
1C38:9B 59 44 41 45 52 9B 20<29>
1C40:DC E4 A5 D4 85 F3 A5 D5<47>
1C48:85 F4 A4 D6 A6 D7 F0 02<A4>
1C50:A0 FF B1 F3 85 97 84 98<FD>
1C58:A9 9B 91 F3 85 92 60 A4<3E>
1C60:98 A5 97 91 F3 A9 00 85<B9>
1C68:92 4C AB 2B A9 06 20 C6<68>
1C70:C2 A2 70 20 04 C6 30 07<C0>
1C78:A2 00 20 98 C2 10 F2 98<D9>
1C80:48 A2 70 20 FD C4 68 C9<E6>
1C88:88 F0 03 4C DC F8 60 A9<CE>
1C90:20 2C A9 23 2C A9 24 2C<C7>
1C98:A9 21 85 C0 A9 07 85 C1<4D>
1CA0:A9 00 A8 4C DC C3 A9 0B<2D>
1CA8:2C A9 07 48 20 A6 C4 20<C7>
1CB0:0C E5 20 AB C4 A5 9B 9D<5D>
1CB8:44 03 A5 9C 9D 45 03 A5<4E>
1CC0:D4 9D 48 03 A5 D5 9D 49<41>
1CC8:03 68 20 1A C5 4C A9 C2<A1>
1CD0:A9 07 85 C0 A0 00 4C 04<C4>
1CD8:C5 20 F4 24 C0 80 B0 01<1C>
1CE0:60 4C A9 C2 A9 0B 2C A9<0F>
1CE8:07 85 C0 20 9C C4 20 6B<76>
1CF0:E5 20 AB C4 A9 D4 85 F3<E6>
1CF8:A9 00 85 F4 A0 06 20 04<CD>
1D00:C5 20 AB 2B 20 B9 C4 A5<DD>
1D08:C0 C9 07 D0 03 20 C0 E5<2F>
1D10:A4 A8 C8 C4 A7 90 D7 60<E0>
1D18:A4 A9 D0 32 C6 AA A6 AA<37>
1D20:C6 AA BD 20 21 20 ED E5<99>
1D28:A0 02 BD 41 21 91 9D C8<AA>
1D30:BD 61 21 91 9D C8 BD 81<5E>
1D38:21 91 9D C8 BD A1 21 91<56>
1D40:9D C8 BD C1 21 91 9D C8<FB>
1D48:BD E1 21 91 9D 60 A9 80<FA>
1D50:85 B1 60 E6 B0 A4 A9 68<CB>
1D58:68 4C 31 E4 A9 40 85 B1<3C>
1D60:24 B1 10 06 A5 AA 85 AF<C3>
1D68:C6 AA A5 B0 A8 F0 07 C6<A0>
1D70:B0 20 16 E5 30 19 84 98<20>
1D78:85 97 20 16 E5 30 10 85<01>
1D80:F5 84 F6 20 6E E5 24 B1<CD>
1D88:50 08 A9 00 85 B1 60 4C<FD>
1D90:2B E5 46 D2 90 F9 A5 F5<0E>
1D98:C5 D6 A5 F6 E5 D7 B0 EF<8F>
1DA0:A5 97 C5 D8 A5 98 E5 D9<17>
1DA8:B0 E5 A5 D9 D0 10 A4 D9<F5>
1DB0:88 D0 0B A5 F5 85 F7 A5<AB>
1DB8:F6 85 F8 4C 14 C7 A0 00<BF>
1DC0:84 F7 84 F8 A0 10 46 D9<13>
1DC8:66 D8 90 0D 18 A5 F7 65<B4>
1DD0:F5 85 F7 A5 F8 65 F6 85<E1>
1DD8:F8 06 F5 26 F6 88 D0 E6<D9>
1DE0:18 A5 97 65 F7 85 F5 A5<D5>
1DE8:98 65 F8 06 F5 2A 85 F6<36>
1DF0:A8 A5 F5 0A 26 F6 65 F5<A9>
1DF8:AA 98 65 F6 A8 8A 65 D4<43>
1E00:AA 98 65 D5 A8 8A 65 8C<D3>
1E08:85 F5 98 65 8D 85 F6 24<B7>
1E10:B1 10 2E A6 AF 86 AA C6<77>
1E18:AA A0 00 84 B1 BD 40 21<79>
1E20:91 F5 C8 BD 60 21 91 F5<5E>
1E28:C8 BD 80 21 91 F5 C8 BD<68>
1E30:A0 21 91 F5 C8 BD C0 21<9B>
1E38:91 F5 C8 BD E0 21 91 F5<6A>
1E40:60 E6 AA A6 AA A0 05 B1<DC>
1E48:F5 9D E0 21 88 B1 F5 9D<3E>
1E50:C0 21 88 B1 F5 9D A0 21<AF>
1E58:88 B1 F5 9D 80 21 88 B1<BE>
1E60:F5 9D 60 21 88 B1 F5 9D<06>
1E68:40 21 5E 00 21 60 A5 B0<5C>
1E70:F0 07 20 0F C8 84 98 85<5F>
1E78:97 20 0F C8 38 E9 01 85<4E>
1E80:F5 98 E9 00 85 F6 20 6E<EC>
1E88:E5 A5 B1 10 0B 05 B0 85<E6>
1E90:B1 A4 D9 A5 D8 4C D0 C7<CA>
1E98:A5 D6 A4 D7 A6 B0 F0 10<6A>
1EA0:C6 B0 C4 98 90 3D D0 04<C2>
1EA8:C5 97 90 37 A4 98 A5 97<08>
1EB0:38 E5 F5 85 D6 AA 98 E5<2C>
1EB8:F6 85 D7 90 26 A8 D0 03<57>
1EC0:8A F0 20 20 DF E4 18 A5<3F>
1EC8:D4 65 F5 85 D4 A5 D5 65<C2>
1ED0:F6 85 D5 24 B1 10 01 60<6D>
1ED8:4C 3A E5 20 16 E5 D0 F7<02>
1EE0:AA D0 F4 A9 05 4C DC F8<87>
1EE8:20 DC E4 A5 D4 85 99 A5<7E>
1EF0:D5 85 9A A5 D6 85 A2 A4<23>
1EF8:D7 84 A3 A4 A9 F0 0F A9<0B>
1F00:80 85 B1 20 31 E4 A5 D7<E9>
1F08:A4 D6 26 B1 B0 07 20 DC<23>
1F10:E4 A5 D9 A4 D8 C5 A3 90<B6>
1F18:06 D0 08 C4 A2 B0 04 85<EB>
1F20:A3 84 A2 18 A5 D4 85 9B<60>
1F28:65 A2 A8 A5 D5 85 9C 65<BD>
1F30:A3 AA 38 98 E5 8C 85 F9<14>
1F38:8A E5 8D 85 FA 20 5D 26<8F>
1F40:A5 D3 20 B3 E4 38 A5 F9<F4>
1F48:E5 D4 A8 A5 FA E5 D5 AA<74>
1F50:A9 02 25 B1 F0 0F A9 00<A0>
1F58:85 B1 E4 D7 90 06 D0 05<B6>
1F60:C4 D6 B0 01 60 84 D6 86<7E>
1F68:D7 4C C0 E5 A4 A8 C4 A7<79>
1F70:90 01 60 20 00 E4 46 D2<13>
1F78:B0 67 38 26 D2 30 65 A5<80>
1F80:F5 69 01 85 F5 85 D6 A5<D6>
1F88:F6 69 00 30 54 85 D7 85<02>
1F90:F6 A5 97 69 01 85 D8 A5<0F>
1F98:98 69 00 85 D9 30 42 A0<F1>
1FA0:00 84 F7 84 F8 A0 10 A5<1A>
1FA8:F5 4A 90 0F 18 A5 F7 65<7D>
1FB0:D8 85 F7 A5 F8 65 D9 85<F1>
1FB8:F8 30 26 66 F8 66 F7 66<B7>
1FC0:F6 66 F5 88 D0 E1 06 F5<7D>
1FC8:26 F6 30 15 A6 F6 A5 F5<31>
1FD0:0A 26 F6 30 0C 65 F5 85<BF>
1FD8:F5 A8 8A 65 F6 85 F6 10<F3>
1FE0:17 4C 2B E5 A9 00 85 D6<7C>
1FE8:85 D7 A4 F5 84 D8 A5 F6<25>
1FF0:85 D9 D0 04 C0 00 F0 E9<30>
1FF8:A2 8E 20 88 25 38 A5 97<09>
2000:E5 8C 85 D4 A5 98 E5 8D<41>
2008:85 D5 20 C0 E5 A9 00 A8<61>
2010:A6 F6 F0 0A 91 97 C8 D0<9F>
2018:FB E6 98 CA D0 F6 A6 F5<85>
2020:F0 06 91 97 C8 CA D0 FA<54>
2028:4C A0 C8 20 C5 C9 A5 8A<47>
2030:85 BE A5 8B 85 BF A5 A1<9A>
2038:AA 0A A8 B1 80 85 8B C8<CB>
2040:B1 80 85 8A A0 01 8A D1<8F>
2048:8A D0 13 88 B1 8A C5 A0<83>
2050:B0 13 A0 02 B1 8A 65 8A<C8>
2058:90 E8 E6 8B B0 E4 90 03<77>
2060:18 90 EF 38 60 D0 FC 18<34>
2068:60 A9 C1 2C A9 C2 AA C8<67>
2070:B1 99 D0 03 C8 B1 99 49<1E>
2078:80 20 ED E5 8A D1 9D F0<87>
2080:33 91 9D A5 99 A0 02 91<88>
2088:9D C8 A5 9A 91 9D 4C E8<CE>
2090:C9 A5 99 48 A5 9A 48 A9<B9>
2098:00 A8 91 80 C8 C8 D0 FA<F5>
20A0:A5 89 85 9A A5 88 85 99<7A>
20A8:A0 04 B1 99 C9 4F F0 B9<79>
20B0:C9 5D F0 B8 A0 01 B1 99<21>
20B8:0A B0 1B A8 B1 80 D0 09<F7>
20C0:A5 9A 91 80 C8 A5 99 91<18>
20C8:80 18 A0 02 B1 99 65 99<B6>
20D0:90 D4 E6 9A B0 D0 A5 88<33>
20D8:85 99 A5 89 85 9A A0 00<18>
20E0:B1 80 D0 0C A5 9A 91 80<C4>
20E8:C8 A5 99 91 80 4C 2B CA<E7>
20F0:85 9A C8 B1 80 85 99 C8<78>
20F8:D0 E6 68 85 9A 68 85 99<EB>
2100:60 A9 1E 4C DC F8 B1 8A<AF>
2108:C8 C4 A7 B0 1A C9 1C D0<FB>
2110:1A E6 A8 20 4F E4 A5 D2<ED>
2118:C9 C2 D0 E5 A0 00 B1 D4<AB>
2120:AA C8 B1 D4 A8 8A 60 A9<69>
2128:00 A8 60 4C 13 E5 20 3A<AB>
2130.CA C0 00 30 09 85 B7 84<96>
2138:B8 A9 00 85 B6 60 4C 28<CC>
2140:E5 A5 B7 85 A0 A5 B8 85<51>
2148:A1 20 62 C9 A5 8A 85 F3<87>
2150:A5 8B 85 F4 A5 BE 85 8A<A9>
2158:A5 BF 85 8B A0 00 B1 F3<A7>
2160:85 B7 C8 B1 F3 85 B8 C8<F8>
2168:B1 F3 85 F5 84 F2 A4 F2<11>
2170:C8 B1 F3 85 F6 C8 84 F2<43>
2178:B1 F3 C9 01 F0 23 A4 F6<42>
2180:C4 F5 B0 03 88 90 E9 88<08>
2188:84 F2 A0 01 B1 F3 30 42<20>
2190:38 A5 F2 65 F3 85 F3 A9<E9>
2198:00 85 B6 65 F4 85 F4 90<76>
21A0:BB A9 00 85 F5 A5 F5 C5<DC>
21A8:B6 B0 12 E6 F2 A4 F2 B1<45>
21B0:F3 C9 9B F0 D5 C9 2C D0<EB>
21B8:F2 E6 F5 D0 E8 A9 40 85<F4>
21C0:A6 E6 F2 4C 42 CB A4 F2<38>
21C8:B1 F3 C9 2C 18 F0 02 C9<CB>
21D0:9B 60 A9 06 4C DC F8 A9<D8>
21D8:3F 85 C2 B1 8A C9 0F D0<8B>
21E0:12 20 86 E4 20 DD CB A4<D0>
21E8:A8 E6 A8 B1 8A C9 12 D0<5F>
21F0:02 66 B4 20 4F E4 C6 A8<42>
21F8:90 05 20 78 DD 85 B4 20<53>
2200:AB 2B 20 7E C2 20 5B CB<58>
2208:A0 00 84 A6 84 F2 20 4F<5A>
2210:E4 E6 A8 A5 D2 30 23 20<04>
2218:C8 2B B0 15 20 FA CA D0<9E>
2220:10 20 C0 E5 4C AC CB A5<0A>
2228:11 F0 01 60 C6 11 4C A4<C4>
2230:F5 A9 00 85 B4 A9 08 4C<05>
2238:DC F8 A0 00 A9 11 8D A2<C3>
2240:20 84 A9 84 B0 84 AA 84<6B>
2248:B1 20 3A E5 C6 F2 A5 F2<34>
2250:85 F5 A2 FF E8 E6 F2 A4<1F>
2258:F2 B1 F3 C9 9B F0 08 C9<2A>
2260:2C D0 F1 24 A6 50 ED A4<FB>
2268:F5 A5 A8 48 8A A2 F3 20<AA>
2270:8B E4 68 85 A8 20 1F C8<84>
2278:24 A6 50 11 E6 B6 A6 A8<6C>
2280:E8 E4 A7 B0 0F 20 FA CA<38>
2288:90 1A 4C BE CA A6 A8 E8<84>
2290:E4 A7 90 08 20 AB 2B A9<D2>
2298:00 85 B4 60 20 FA CA 90<C7>
```

Listing »Turbo-Basic XL« (Fortsetzung)



```
22A0:03 4C 33 CB E6 F2 4C 42<5C>
22A8:CB A2 00 A5 D4 9D 44 03<E1>
22B0:A5 D5 9D 45 03 A5 D6 9D<9B>
22B8:48 03 A5 D7 9D 49 03 A9<3C>
22C0:0B 4C FE C5 00 D8 F9 DF<DA>
22C8:20 05 E5 85 A2 84 A3 4C<F6>
22D0:5D 26 20 05 E5 85 A2 84<A9>
22D8:A3 4C DE 25 20 9B E5 20<44>
22E0:9D 2C 4C 36 E5 20 9B E5<8E>
22E8:20 97 2C 4C 36 E5 20 9B<4C>
22F0:E5 20 9E 26 4C 36 E5 20<E9>
22F8:9B E5 20 00 28 4C 36 E5<82>
2300:A6 AA BD 40 21 F0 05 49<53>
2308:80 9D 40 21 60 20 DC D8<9A>
2310:90 62 F0 60 B0 50 20 DC<95>
2318:D8 F0 4B D0 57 20 DC D8<77>
2320:90 52 B0 42 20 DC D8 90<6C>
2328:3D F0 3B B0 47 20 DC D8<35>
2330:90 34 B0 40 20 DC D8 F0<15>
2338:3B D0 2B C6 AA A6 AA C6<56>
2340:AA BD 40 21 3D 41 21 0A<5C>
2348:F0 1C D0 28 C6 AA A6 AA<D8>
2350:C6 AA BD 40 21 1D 41 21<2F>
2358:0A F0 0B D0 17 A6 AA C6<56>
2360:AA BD 40 21 F0 0E A9 00<4E>
2368:A8 F0 0D A0 03 2C A0 02<54>
2370:A9 40 D0 04 A9 40 A0 01<DB>
2378:E6 AA A6 AA 9D 40 21 98<3B>
2380:9D 60 21 A9 00 9D 00 21<7F>
2388:9D 80 21 9D A0 21 9D C0<97>
2390:21 9D E0 21 60 A6 AA C6<51>
2398:AA BD 40 21 0A F0 C7 A9<43>
23A0:80 6A D0 D2 A4 A9 B9 A3<16>
23A8:20 C9 2F 90 03 4C 2E D9<8E>
23B0:A4 AA C6 AA A6 AA C6 AA<D0>
23B8:BD 40 21 D9 40 21 D0 2C<79>
23C0:0A 90 02 E8 88 BD 60 21<FE>
23C8:D9 60 21 D0 1E BD 80 21<FB>
23D0:D9 80 21 D0 16 BD A0 21<CB>
23D8:D9 A0 21 D0 0E BD C0 21<DB>
23E0:D9 C0 21 D0 06 BD E0 21<A7>
23E8:D9 E0 21 60 19 40 21 10<D8>
23F0:FA 6A 49 80 2A 60 20 DC<A5>
23F8:E4 20 9C 2E 20 DC E4 A0<66>
2400:00 A5 D6 D0 06 A5 D7 F0<CB>
2408:05 C6 D7 C6 D6 AA 08 A5<A6>
2410:E2 D0 06 A5 E3 F0 20 C6<6D>
2418:E3 C6 E2 28 F0 16 B1 D4<1F>
2420:D1 E0 D0 0E E6 D4 D0 02<C5>
2428:E6 D5 E6 E0 D0 D3 E6 E1<C9>
2430:D0 CF B0 02 C8 18 60 28<B1>
2438:38 60 20 DC E4 A5 D6 A4<79>
2440:D7 85 D4 84 D5 20 DC 2B<6A>
2448:A9 00 85 D2 85 D3 4C 3A<7D>
2450:E5 20 16 E5 A0 00 B1 D4<56>
2458:4C 79 D9 C6 AA 38 AD E5<19>
2460:02 E5 90 AA AD E6 02 E5<85>
2468:91 A8 8A 4C 79 D9 20 73<5B>
2470:C5 A9 00 85 F2 20 C8 2B<68>
2478:20 93 C5 90 CB 4C D7 F8<74>
2480:20 DC E4 A0 00 B1 D4 4C<FC>
2488:79 D9 20 73 C5 A0 00 84<AB>
2490:F2 84 D4 84 D5 20 68 EA<87>
2498:90 FB 20 93 C5 4C 7D D9<BD>
24A0:20 DC E4 4C 7D D9 A9 00<C3>
24A8:F0 0A A9 08 D0 06 A9 0C<FC>
24B0:D0 02 A9 14 85 EC 20 16<01>
24B8:E5 D0 0F C9 08 B0 0B 65<CF>
24C0:EC AA BD 70 02 A0 00 4C<62>
24C8:79 D9 4C 28 E5 20 AB 2B<37>
24D0:A0 00 A5 D5 F0 03 20 13<47>
24D8:DA A5 D4 48 4A 4A 4A 4A<CB>
24E0:20 1E DA 68 29 0F 09 30<39>
24E8:C9 3A 90 02 69 06 91 F3<C6>
24F0:C8 60 20 16 E5 20 05 DA<25>
24F8:A9 80 D0 3B 20 6E E5 20<F2>
2500:BE 2B A0 FF C8 B1 F3 10<CE>
2508:FB 29 7F 91 F3 C8 A5 F3<E5>
2510:D0 25 AD FC 02 A0 00 C9<3C>
2518:C0 B0 1A A2 0E DD 93 DC<C4>
2520:F0 13 CA 10 F8 20 0D C6<81>
2528:4C 68 DA 20 16 E5 A5 D4<CC>
2530:8D C0 05 A0 01 A9 C0 A2<69>
2538:05 86 D5 85 D4 84 D6 A9<A0>
2540:00 85 D7 85 D3 A9 83 85<E4>
2548:D2 4C 3A E5 C6 AA A9 3F<C9>
2550:85 D4 A2 05 AD 0A D2 29<31>
2558:F0 C9 A0 B0 F7 85 E0 AD<02>
2560:0A D2 29 0F C9 0A B0 F7<FF>
2568:05 E0 95 D4 CA D0 E5 20<94>
2570:B6 2D 4C 80 D9 20 86 DA<91>
2578:20 26 D8 4C FF DA A6 AA<74>
2580:BD 40 21 29 7F 9D 40 21<F7>
2588:60 20 6E E5 A5 D4 29 7F<EC>
2590:38 E9 40 90 14 AA A9 00<CA>
2598:E0 05 90 02 A2 04 95 D5<41>
25A0:CA 10 FB 20 B8 2D 4C 3A<78>
25A8:E5 E6 AA 60 20 2F D8 20<A7>
25B0:6E E5 A5 D4 29 7F 38 E9<79>
25B8:40 90 49 AA A9 00 E0 04<DA>
25C0:B0 3D 95 D6 E8 90 F7 20<31>
25C8:6E E5 A5 D4 29 7F 38 E9<13>
25D0:40 90 2F AA E0 04 B0 D1<CD>
25D8:A9 00 A8 15 D6 94 D6 E8<CF>
25E0:E0 04 D0 F7 24 D4 10 17<2C>
25E8:A8 F0 14 A9 C0 85 E0 A0<92>
25F0:01 84 E1 88 84 E2 84 E3<C9>
25F8:84 E4 84 E5 20 9D 2C 4C<17>
2600:3A E5 06 D4 20 9C 2B 90<F6>
2608:F6 B0 E0 20 6E E5 20 80<40>
2610:30 4C 38 E5 20 6E E5 20<41>
2618:8A 30 4C 38 E5 20 6E E5<99>
2620:20 48 31 4C 38 E5 20 6E<3C>
2628:E5 20 EE 2F 4C 38 E5 20<C4>
2630:6E E5 20 F2 2F 4C 38 E5<97>
2638:20 6E E5 20 44 2F 4C 38<74>
2640:E5 20 6E E5 20 C5 31 4C<29>
2648:38 E5 4C 9E D8 4C AC D8<7C>
2650:A5 E0 10 F6 4C 28 E5 20<B5>
2658:9B E5 A5 E0 F0 EF A5 D4<A6>
2660:F0 EE 10 26 29 7F 85 D4<A8>
2668:A5 E0 29 7F 38 E9 40 90<B0>
2670:E3 A2 04 C9 04 F0 0E B0<4F>
2678:11 AA A8 B9 E2 00 D0 D4<03>
2680:C8 C0 04 D0 F6 B5 E1 4A<BC>
2688:B0 01 18 A9 00 6A 48 A5<55>
2690:E0 C9 40 D0 4D A5 E2 05<E9>
2698:E3 05 E4 05 E5 D0 43 A5<F6>
26A0:E1 29 F0 4A 85 FC 4A 4A<44>
26A8:65 FC 85 FC A5 E1 29 0F<A0>
26B0:65 FC 85 FC 20 85 2E 46<14>
26B8:FC B0 03 20 BB 2F 20 D4<2F>
26C0:2E A5 FC F0 50 20 F3 2E<E1>
26C8:20 9B 26 B0 0F 46 FC 90<A0>
26D0:F7 20 B5 2E 20 23 2F 20<25>
26D8:9E 26 90 E2 4C 2E E5 4C<3B>
26E0:28 E5 A5 E5 48 A5 E4 48<81>
26E8:A5 E3 48 A5 E2 48 A5 E1<0D>
26F0:48 A5 E0 48 20 F2 2F B0<E1>
26F8:E6 68 85 E0 68 85 E1 68<26>
2700:85 E2 68 85 E3 68 85 E4<14>
2708:68 85 E5 20 9E 26 B0 CC<C2>
2710:20 59 2F B0 C7 68 10 04<5D>
2718:05 D4 85 D4 4C 3A E5 A9<69>
2720:06 2C A9 00 85 FB 60 20<88>
2728:16 E5 85 E0 84 E1 4C 16<5F>
2730:E5 20 5F DC 98 25 E1 A8<45>
2738:A5 E0 25 D4 4C 79 D9 20<4D>
2740:5F DC 98 05 E1 A8 A5 E0<51>
2748:05 D4 4C 79 D9 20 5F DC<11>
2750:98 45 E1 A8 A5 E0 45 D4<DC>
2758:4C 79 D9 9A 98 9D 9B B3<0B>
2760:B5 B0 B2 A6 3C 7C BC 27<34>
2768:67 A7 20 13 E5 A5 14 A4<D3>
2770:11 F0 11 C5 14 F0 F8 A5<A7>
2778:D4 D0 04 C6 D5 30 05 C6<B8>
2780:D4 4C A5 DC 60 A6 AA CA<A0>
2788:E6 AA E6 AA 18 BD 00 21<B4>
2790:9D 02 21 BD 01 21 9D 03<78>
2798:21 8A 69 20 AA 90 EE 60<B8>
27A0:20 BD DC 20 E4 DA 20 26<98>
27A8:D8 4C 1D D8 A5 C3 4C FD<E0>
27B0:D9 A5 BA A4 BB 4C 79 D9<D4>
27B8:E6 C9 A5 C9 F0 FA 85 AF<FE>
27C0:A9 00 85 94 A4 A8 B1 8A<85>
27C8:C9 12 F0 4C C9 16 F0 7A<2A>
27D0:C9 14 F0 76 C9 15 F0 6D<D4>
27D8:C9 1C F0 5A 20 00 E4 C6<B5>
27E0:A8 A6 AA BD 00 21 30 03<A4>
27E8:20 34 DA 20 DC E4 A6 B5<F7>
27F0:20 AF C2 A5 D4 9D 44 03<67>
27F8:A5 D5 9D 45 03 A5 D6 9D<A9>
2800:48 03 18 65 94 85 94 A5<09>
2808:D7 9D 49 03 05 D6 F0 B4<99>
2810:A9 0B 20 FE C5 4C FC DC<86>
2818:A4 94 C8 C4 AF 90 09 18<32>
2820:A5 C9 65 AF 85 AF 90 F0<AE>
2828:A4 94 C4 AF B0 17 20 02<EE>
2830:F2 E6 94 4C 60 DD 20 78<D9>
2838:DD 85 B5 C6 A8 4C FC DC<CD>
2840:E6 A8 4C 13 E5 E6 A8 4C<CA>
2848:FC DC A4 A8 88 B1 8A C9<6B>
2850:15 F0 07 C9 12 F0 03 20<0A>
2858:94 C2 A9 00 85 B5 60 A9<03>
2860:13 85 F3 A9 23 85 F4 C6<21>
2868:DB A2 07 86 B5 A9 00 A0<B0>
2870:08 20 B4 C3 20 B9 C4 20<06>
2878:F2 DC 4C F8 C4 A9 5F 2C<91>
2880:A9 FF 85 DF A4 B0 88 98<0A>
2888:F0 03 20 16 E5 85 DA 84<69>
2890:DB 20 DC E4 20 9C 2E 20<FF>
2898:DC E4 18 A5 D4 65 DA 85<96>
28A0:D4 A5 D5 65 DB 85 D5 B0<47>
28A8:22 38 A5 D6 E5 DA 85 D6<F9>
28B0:A5 D7 E5 DB 85 D7 90 13<8E>
28B8:05 D6 F0 0F 38 A5 D6 E5<F0>
28C0:E2 85 DC A5 D7 E5 E3 85<E5>
28C8:DD B0 03 4C 9E D8 E6 DA<CB>
28D0:D0 02 E6 DB A0 00 B1 E0<39>
28D8:51 D4 25 DF D0 30 A5 D4<BC>
28E0:85 D8 A5 D5 85 D9 A5 E0<A5>
28E8:85 E4 A5 E1 85 E5 A5 E3<7C>
28F0:85 DE A6 E2 E8 CA D0 04<1B>
28F8:C6 DE 30 25 B1 D8 51 E4<68>
2900:25 DF D0 0A C8 D0 EE E6<76>
2908:D9 E6 E5 4C 2D DE E6 D4<32>
2910:D0 02 E6 D5 A5 DC D0 04<0D>
2918:C6 DD 30 AF C6 DC 4C 06<B1>
2920:DE A5 DA A4 DB 4C 79 D9<D7>
2928:85 D6 85 D7 A0 08 16 E6<67>
2930:36 E5 26 D6 90 0B 18 B5<8E>
2938:E6 65 D7 95 E6 90 02 F6<B2>
2940:E5 88 D0 EA 60 4C 28 E5<2F>
2948:20 05 E5 D0 F8 48 A4 A8<57>
2950:C8 C4 A7 B0 05 20 13 E5<37>
2958:D0 EB 68 D0 02 A9 01 85<91>
2960:D5 A2 16 A9 00 95 D6 CA<9A>
2968:10 FB A5 D5 85 E0 E8 20<1E>
2970:60 DE A5 D4 A2 02 20 60<11>
2978:DE A5 D5 85 D6 A0 08 06<2D>
2980:DC 26 DB 26 DA 06 D6 90<32>
2988:11 18 A5 DC 65 E8 85 DC<8D>
2990:A5 DB 65 E7 85 DB 90 02<B9>
2998:E6 DA 88 D0 E2 20 37 DF<1D>
29A0:20 5C DF 20 37 DF 20 6A<4E>
29A8:DF 20 4E DF 20 5C DF 20<76>
29B0:4E DF 20 6A DF 24 E2 30<06>
29B8:21 E6 E1 18 A5 EB 65 E6<3B>
29C0:85 EB A5 EA 65 E5 85 EA<74>
29C8:90 02 E6 E9 38 A2 02 B5<4B>
29D0:E2 F5 E9 95 E2 CA 10 F7<04>
29D8:30 C3 A5 E0 F0 2F C6 E0<6F>
29E0:38 A5 DC E5 E8 85 DC A5<33>
29E8:DB E5 E7 85 DB B0 02 C6<89>
29F0:DA 18 A2 02 B5 E2 75 DA<83>
29F8:95 E2 CA 10 F7 30 9E 18<12>
2A00:A5 99 65 E0 85 55 A5 9A<D0>
2A08:69 00 85 56 60 A2 00 86<8E>
2A10:54 86 55 86 56 60 38 A5<FC>
2A18:99 E5 E0 85 55 A5 9A E9<8B>
2A20:00 85 56 60 18 A5 9B 65<33>
2A28:E1 85 54 A5 9C 69 00 F0<59>
2A30:0E 60 38 A5 9B E5 E1 85<A4>
2A38:54 A5 9C E9 00 D0 F2 A4<A9>
2A40:C8 A2 60 20 DE 24 C0 80<EE>
2A48:D0 E7 4C A9 C2 84 96 A2<61>
2A50:00 86 AF 85 95 A6 F2 A0<07>
2A58:00 B1 95 F0 27 BD 80 05<9C>
2A60:29 7F E8 51 95 D0 03 C8<F5>
2A68:D0 F3 0A F0 15 B0 06 C8<DD>
2A70:B1 95 10 FB 38 E6 AF F0<9A>
2A78:0D 98 65 95 90 D5 E6 96<90>
2A80:B0 D1 18 60 38 60 A9 04<7E>
2A88:4C DC F8 A2 00 86 AF 84<CA>
2A90:96 85 95 A6 F2 A0 01 B1<92>
2A98:95 F0 E9 BD 80 05 29 7F<0D>
2AA0:E8 C9 2E F0 DD 51 95 D0<F5>
2AA8:03 C8 D0 EF 0A F0 D3 B0<BF>
2AB0:05 C8 B1 95 10 FB E6 AF<FF>
2AB8:38 98 65 95 90 D3 E6 96<31>
2AC0:B0 CF 00 E4 F1 FF A0 00<31>
2AC8:A9 11 8D A2 20 84 A9 84<C9>
2AD0:B0 84 AA 84 B1 20 4F E4<63>
2AD8:B0 06 20 3A E5 4C 0F E4<87>
2AE0:85 AB AA BD CC 23 85 AC<62>
2AE8:A4 A9 BE A2 20 BD 2A 24<C6>
2AF0:C5 AC 90 0E AA F0 16 B9<56>
2AF8:A2 20 C6 A9 20 48 E4 4C<BE>
2B00:22 E4 A5 AB C8 99 A2 20<D5>
2B08:84 A9 4C 0F E4 60 0A 8D<02>
2B10:4D E4 6C 2F 23 A4 A8 E6<E2>
2B18:A8 B1 8A 30 5A F0 53 C9<F3>
2B20:0F 90 03 F0 27 60 C8 B1<3C>
2B28:8A 85 D4 C8 B1 8A 85 D5<49>
2B30:C8 B1 8A 85 D6 C8 B1 8A<53>
2B38:85 D7 C8 B1 8A 85 D8 C8<F9>
2B40:B1 8A 85 D9 C8 84 A8 A9<46>
2B48:00 85 D2 60 C8 B1 8A A2<EE>
2B50:8A 85 D6 85 D8 C8 98 18<3D>
2B58:75 00 85 D4 A9 00 85 D7<7E>
2B60:85 D9 75 01 85 D5 98 65<7A>
2B68:D6 A8 A9 83 85 D2 84 A8<F5>
```

```
2B70:18 60 C8 E6 A8 B1 8A 49<A8>
2B78:80 85 D3 20 ED E5 B1 9D<45>
2B80:85 D2 A0 02 B1 9D 85 D4<7E>
2B88:C8 B1 9D 85 D5 C8 B1 9D<DF>
2B90:85 D6 C8 B1 9D 85 D7 C8<17>
2B98:B1 9D 85 D8 C8 B1 9D 85<9C>
2BA0:D9 60 20 6E E5 A9 02 24<48>
2BA8:D2 D0 15 05 D2 85 D2 4A<61>
2BB0:90 3F 18 A5 D4 65 8C 85<2B>
2BB8:D4 A8 A5 D5 65 8D 85 D5<9F>
2BC0:60 20 13 E5 10 FA A9 07<CF>
2BC8:4C DC F8 20 13 E5 85 99<4C>
2BD0:84 9A 20 13 E5 85 9B 84<F6>
2BD8:9C 20 00 E4 20 6E E5 20<23>
2BE0:25 33 B0 0A A5 D4 A4 D5<BE>
2BE8:60 20 13 E5 F0 D2 A9 03<42>
2BF0:2C A9 09 2C A9 0B 2C A9<E3>
2BF8:0A 4C DC F8 B0 F6 B0 EE<0C>
2C00:E6 AA A4 AA C0 20 B0 EF<D1>
2C08:A5 D9 99 E0 21 A5 D8 99<FE>
2C10:C0 21 A5 D7 99 A0 21 A5<C3>
2C18:D6 99 80 21 A5 D5 99 60<AF>
2C20:21 A5 D4 99 40 21 A5 D3<96>
2C28:99 20 21 A5 D2 99 00 21<D5>
2C30:60 20 00 E4 A4 AA C6 AA<C8>
2C38:B9 E0 21 85 D9 B9 C0 21<1F>
2C40:85 D8 B9 A0 21 85 D7 B9<E4>
2C48:80 21 85 D6 B9 60 21 85<45>
2C50:D5 B9 40 21 85 D4 B9 20<68>
2C58:21 85 D3 B9 00 21 85 D2<08>
2C60:60 C6 AA A4 AA B9 E1 21<E6>
2C68:85 E5 B9 C1 21 85 E4 B9<47>
2C70:A1 21 85 E3 B9 81 21 85<D7>
2C78:E2 B9 61 21 85 E1 B9 41<CA>
2C80:21 85 E0 4C 70 E5 A5 D3<80>
2C88:20 ED E5 A5 D2 91 9D C8<E9>
2C90:A5 D3 91 9D C8 A5 D4 91<E5>
2C98:9D C8 A5 D5 91 9D C8 A5<48>
2CA0:D6 91 9D C8 A5 D7 91 9D<80>
2CA8:C8 A5 D8 91 9D C8 A5 D9<BF>
2CB0:91 9D 60 0A 2A 2A 2A A8<BB>
2CB8:6A 29 F8 18 65 86 85 9D<E7>
2CC0:98 29 07 65 87 85 9E A0<F9>
2CC8:00 60 A9 00 85 92 85 CA<8E>
2CD0:AD E7 02 AC E8 02 85 80<84>
2CD8:84 81 C8 85 82 84 83 85<7F>
2CE0:84 84 85 18 69 01 90 01<D6>
2CE8:C8 85 86 84 87 85 88 84<39>
2CF0:89 85 8A 84 8B 18 69 03<FD>
2CF8:90 01 C8 85 8C 84 8D 85<BC>
2D00:8E 84 8F 85 90 84 91 85<82>
2D08:0E 84 0F A9 00 A8 91 84<FA>
2D10:91 8A C8 A9 80 91 8A C8<BF>
2D18:A9 03 91 8A A9 0A 85 C9<99>
2D20:4C 66 FF A2 FF 9A D8 A5<68>
2D28:CA F0 03 20 04 E6 20 E7<23>
2D30:F6 20 34 C5 20 D1 F5 A5<33>
2D38:92 F0 03 20 93 C5 20 5C<84>
2D40:C5 20 CB C0 A5 CA D0 E3<91>
2D48:A2 FF 9A D8 20 AB 2B A9<0F>
2D50:5D 85 C2 20 87 C2 A5 11<8C>
2D58:D0 04 C6 11 D0 E6 A0 00<BA>
2D60:84 F2 84 9F 84 94 84 A6<F5>
2D68:84 B3 84 B0 84 B1 A5 84<9D>
2D70:85 AD A5 85 85 AE 20 1D<8D>
2D78:EB 20 FB E7 20 F7 E8 A5<6F>
2D80:D5 10 02 85 A6 20 1D EB<AC>
2D88:84 A8 B1 F3 C9 9B D0 07<2B>
2D90:24 A6 30 B0 4C E0 E7 A5<B4>
2D98:94 85 A7 20 F7 E8 20 1D<69>
2DA0:EB A0 EB A9 5A 20 C3 DF<F8>
2DA8:66 EC 30 08 A5 AF C9 15<3D>
2DB0:D0 23 86 DA A0 EC A9 80<11>
2DB8:20 C3 DF A9 36 B0 06 65<1A>
2DC0:AF 69 02 90 10 24 EC 30<5E>
2DC8:0C A9 BF 85 95 A9 EB 85<1C>
2DD0:96 A9 15 A6 DA 86 F2 20<75>
2DD8:F7 E8 20 B4 2B 20 21 E8<E2>
2DE0:90 35 A4 9F B1 F3 C9 9B<3B>
2DE8:D0 06 C8 91 F3 88 A9 20<E8>
2DF0:09 80 91 F3 A9 40 05 A6<2E>
2DF8:85 A6 A4 A8 84 F2 A2 03<67>
2E00:86 A7 E8 86 94 A9 37 20<FB>
2E08:F7 E8 A4 F2 B1 F3 E6 F2<7C>
2E10:C9 9B D0 F3 20 F7 E8 A5<32>
2E18:94 A4 A7 99 00 21 A4 F2<65>
2E20:88 B1 F3 C9 9B F0 03 4C<7E>
2E28:D1 E6 A0 02 A5 94 99 00<C5>
2E30:21 20 5F C9 A9 00 B0 04<56>
2E38:A0 02 B1 8A 38 E5 94 F0<A0>
2E40:24 B0 13 49 FF A8 C8 A2<4B>
2E48:8A 20 86 25 A5 97 85 8A<09>
2E50:A5 98 85 8B D0 0F A8 18<F0>
2E58:65 8A 85 8A 90 02 E6 8B<F6>
2E60:A2 8A 20 20 26 A4 94 88<A5>
2E68:B9 00 21 91 8A 98 D0 F7<9F>
2E70:24 A6 50 2A A5 B1 0A 0A<FA>
2E78:0A A8 A2 88 20 20 26 38<96>
2E80:A5 84 E5 AD A8 A5 85 E5<39>
2E88:AE A2 84 20 22 26 24 A6<4A>
2E90:10 06 20 31 F2 4C 7E E6<31>
2E98:20 07 F2 4C 7E E6 10 FB<EF>
2EA0:20 26 C1 4C AF FF 20 5F<F3>
2EA8:C9 B0 F0 A0 02 B1 8A A8<6E>
2EB0:18 65 8A 85 8A 90 02 E6<77>
2EB8:8B A2 8A 20 20 26 4C 7E<45>
2EC0:E6 20 C8 2B 90 08 A9 00<B8>
2EC8:85 F2 A0 80 30 0B 20 25<8A>
2ED0:33 B0 F3 A4 D5 30 EF A5<86>
2ED8:D4 84 A1 85 A0 20 F7 E8<FF>
2EE0:A5 A1 85 D5 4C F7 E8 A0<82>
2EE8:00 84 A9 B1 95 0A A8 B9<7C>
2EF0:39 F0 85 9D 8D 82 04 B9<9A>
2EF8:3A F0 85 9E 8D 83 04 A5<17>
2F00:94 8D 81 04 A5 F2 8D 80<43>
2F08:04 E6 9D D0 02 E6 9E A2<19>
2F10:00 A1 9D 30 0C C9 05 90<4A>
2F18:4F 20 E6 E8 90 EB 4C BC<65>
2F20:E8 0A A8 B9 3A F0 48 B9<50>
2F28:39 F0 48 C0 12 B0 0F 68<DD>
2F30:A8 68 20 74 E8 90 D2 4C<9D>
2F38:BC E8 48 98 48 60 A6 A9<25>
2F40:E8 E8 E8 E8 F0 1F 86 A9<BE>
```

Listing »Turbo-Basic XL« (Fortsetzung)



```
2F48:A5 F2 9D 80 04 A5 94 9D<FA>
2F50:81 04 A5 9D 9D 82 04 A5<76>
2F58:9E 9D 83 04 68 85 9D 68<90>
2F60:85 9E 4C 43 E8 4C 00 E9<BF>
2F68:A6 A9 D0 01 60 BD 82 04<E6>
2F70:85 9D BD 83 04 85 9E CA<85>
2F78:CA CA CA 86 A9 B0 03 4C<29>
2F80:43 E8 E6 9D D0 02 E6 9E<B9>
2F88:A2 00 A1 9D 30 F4 C9 03<7B>
2F90:F0 D6 B0 EE A5 F2 C5 9F<50>
2F98:90 02 85 9F A6 A9 BD 80<8B>
2FA0:04 85 F2 BD 81 04 85 94<AF>
2FA8:4C 43 E8 60 C9 0F D0 41<63>
2FB0:E6 9D D0 02 E6 9E A2 00<6D>
2FB8:A1 9D 18 C6 94 A4 94 99<6A>
2FC0:00 21 E6 94 D0 E5 A9 0E<DF>
2FC8:4C DC F8 A2 FF 9A A5 94<AC>
2FD0:A4 A7 99 00 21 4C D1 E6<DE>
2FD8:A2 FF 9A A0 04 B9 00 21<3C>
2FE0:C9 54 D0 0A A4 F2 88 A9<F7>
2FE8:9B 91 F3 4C 51 E7 4C 44<1D>
2FF0:E7 20 1D EB C4 B3 F0 14<F8>
2FF8:84 B3 A0 ED A9 71 20 85<77>
3000:DF B0 2C 86 B2 18 A5 AF<B8>
3008:69 10 85 B0 A0 00 B1 9D<57>
3010:C5 B0 F0 12 C9 44 D0 0A<1F>
3018:A5 B0 C9 56 B0 04 C9 44<D8>
3020:B0 02 38 60 A5 B0 20 F7<61>
3028:E8 A6 B2 86 F2 18 60 A9<6E>
3030:00 85 B0 38 60 A9 C0 D0<60>
3038:06 A9 00 F0 02 A9 80 85<FF>
3040:D2 20 1D EB 84 AC 20 53<45>
3048:EA B0 28 20 2B E9 A5 B0<76>
3050:F0 08 A4 B2 B1 F3 C9 30<75>
3058:90 19 E6 F2 20 53 EA 90<50>
3060:F9 C9 30 90 04 C9 3A 90<C4>
3068:F1 C9 24 F0 08 24 D2 10<B9>
3070:0D 70 18 38 60 24 D2 10<41>
3078:FA 70 F8 C8 D0 0D B1 F3<82>
3080:C9 28 D0 07 C8 A9 40 05<28>
3088:D2 85 D2 A5 AC 85 F2 84<93>
3090:AC A4 83 A5 82 20 85 DF<17>
3098:B0 0A E4 AC F0 41 20 AC<FB>
30A0:DF 4C D2 E9 38 A5 AC E5<3E>
30A8:F2 85 F2 A8 A2 84 20 86<10>
30B0:25 A5 AF 85 D3 A4 F2 88<8F>
30B8:A6 AC CA BD 80 05 09 80<05>
30C0:91 97 CA BD 80 05 88 10<CA>
30C8:F7 A0 08 A2 88 20 86 25<93>
30D0:E6 B1 20 9C 2B A0 07 B9<BC>
30D8:D2 00 91 97 88 10 F8 98<D1>
30E0:48 A5 9D 48 A6 9E A5 AF<41>
30E8:20 ED E5 B1 9D 45 D2 A8<90>
30F0:86 9E 68 85 9D C0 80 68<4E>
30F8:A8 B0 A3 24 D2 50 04 30<B6>
3100:02 C6 AC A5 AC 85 F2 A5<8E>
3108:AF 10 07 A9 00 20 F7 E8<87>
3110:A5 AF 49 80 20 F7 E8 18<96>
3118:60 A4 F2 B1 F3 C9 5F F0<D9>
3120:F6 C9 41 90 07 C9 5B 60<C3>
3128:A4 AC 84 F2 38 60 A4 F2<ED>
3130:B1 F3 38 E9 30 90 F5 C9<30>
3138:0A 90 0A C9 11 90 ED E9<DC>
3140:07 C9 10 B0 E7 A0 04 06<1D>
3148:D4 26 D5 88 D0 F9 05 D4<79>
3150:85 D4 E6 F2 18 60 20 1D<E1>
3158:EB 84 AC B1 F3 C9 24 F0<84>
3160:07 20 C8 2B 90 20 B0 C0<1E>
3168:E6 F2 20 1D EB 20 9C 2B<42>
3170:20 68 EA B0 B3 20 68 EA<04>
3178:B0 06 A5 D5 29 F0 F0 F5<F3>
3180:20 DC 2B A9 0D 2C A9 0E<52>
3188:20 F7 E8 A4 94 A2 00 B5<DC>
3190:D4 99 00 21 C8 E8 E0 06<B8>
3198:90 F5 84 94 18 60 20 1D<75>
31A0:EB B1 F3 C9 22 F0 02 38<27>
31A8:60 A9 0F 20 F7 E8 A5 94<80>
31B0:85 AB 20 F7 E8 E6 F2 A4<71>
31B8:F2 B1 F3 C9 9B F0 0D C9<EA>
31C0:22 D0 EF E6 F2 C8 B1 F3<68>
31C8:C9 22 F0 E6 18 A5 94 E5<D8>
31D0:AB A4 AB 99 00 21 18 60<B0>
31D8:AD 01 21 0A B0 04 A5 94<00>
31E0:C9 06 60 A4 F2 B1 F3 29<DF>
31E8:7F C9 20 D0 03 C8 D0 F5<CA>
31F0:84 F2 B1 F3 C9 9B F0 25<8F>
31F8:29 7F 91 F3 C9 61 90 08<D2>
3200:C9 7B B0 04 29 5F 91 F3<D6>
3208:C8 C9 30 90 10 C9 3A 90<29>
3210:E1 C9 41 90 08 C9 5B 90<E4>
3218:D9 C9 5F F0 D5 A4 F2 60<63>
3220:95 52 45 CD 95 44 41 54<ED>
3228:C1 98 49 4E 50 55 D4 91<5D>
3230:43 4F 4C 4F D2 9D 4C 49<32>
3238:53 D4 96 45 4E 54 45 D2<92>
3240:9A 4C 45 D4 9B 49 C6 9C<D0>
3248:46 4F D2 97 4E 45 58 D4<6B>
3250:91 47 4F 54 CF 91 47 4F<D8>
3258:20 54 CF 91 47 4F 53 55<53>
3260:C2 BD 54 52 41 D0 90 42<1E>
3268:59 C5 90 43 4F 4E D4 9E<D6>
3270:43 4F CD 9F 43 4C 4F 53<D6>
3278:C5 90 43 4C D2 90 44 45<E2>
3280:C7 9E 44 49 CD 90 45 4E<88>
3288:C4 90 4E 45 D7 A0 4F 50<2C>
3290:45 CE 96 4C 4F 41 C4 96<AF>
3298:53 41 56 C5 A2 53 54 41<BF>
32A0:54 55 D3 A3 4E 4F 54 C5<FB>
32A8:CF 50 4F 49 4E D4 A1 58<9B>
32B0:49 CF A4 4F CE 92 50 4F<7E>
32B8:4B C5 A5 50 52 49 4E D4<F1>
32C0:90 52 41 C4 99 52 45 41<FE>
32C8:C4 A8 52 45 53 54 4F 52<8B>
32D0:C5 90 52 45 54 55 52 CE<9F>
32D8:A9 52 55 CE 90 53 54 4F<6E>
32E0:D0 90 50 4F D0 A5 BF 98<37>
32E8:47 45 D4 AB 50 55 D4 91<50>
32F0:47 52 41 50 48 49 43 D3<36>
32F8:92 50 4C 4F D4 92 50 4F<6C>
3300:53 49 54 49 4F CE 90 44<07>
3308:4F D3 92 44 52 41 57 54<67>
3310:CF 93 53 45 54 43 4F 4C<C7>
3318:4F D2 AC 4C 4F 43 41 54<A0>
3320:C5 94 53 4F 55 4E C4 A6<B7>
3328:4C 50 52 49 4E D4 90 43<83>
3330:53 41 56 C5 90 43 4C 4F<18>
3338:41 C4 9A 00 80 00 45 52<43>
3340:52 4F 52 2D 20 A0 92 44<99>
3348:50 4F 4B C5 93 4D 4F 56<26>
3350:C5 93 2D 4D 4F 56 C5 AA<5D>
3358:2A C6 90 52 45 50 45 41<B1>
3360:D4 91 55 4E 54 49 CC 91<42>
3368:57 48 49 4C C5 90 57 45<96>
3370:4E C4 90 45 4C 53 C5 90<FA>
3378:45 4E 44 49 C6 CF 42 50<E7>
3380:55 D4 CF 42 47 45 D4 92<9D>
3388:46 49 4C 4C 54 CF 90 44<65>
3390:CF 90 4C 4F 4F D0 A8 45<00>
3398:58 49 D4 DA 44 49 D2 96<93>
33A0:4C 4F 43 CB 96 55 4E 4C<DF>
33A8:4F 43 CB 96 52 45 4E 41<9A>
33B0:4D C5 96 44 45 4C 45 54<1A>
33B8:C5 91 50 41 55 53 C5 96<0A>
33C0:54 49 4D 45 24 BD A7 50<43>
33C8:52 4F C3 B8 45 58 45 C3<E0>
33D0:90 45 4E 44 50 52 4F C3<C2>
33D8:91 46 43 4F 4C 4F D2 AA<3E>
33E0:2A CC 95 2D AD 93 52 45<4F>
33E8:4E 55 CD 92 44 45 CC DA<F2>
33F0:44 55 4D D0 AA 54 52 41<B8>
33F8:43 C5 DB 54 45 58 D4 96<86>
3400:42 4C 4F 41 C4 96 42 52<48>
3408:55 CE B8 47 4F A3 A7 A3<70>
3410:AA 2A C2 92 50 41 49 4E<9D>
3418:D4 9F 43 4C D3 94 44 53<80>
3420:4F 55 4E C4 DC 43 49 52<1B>
3428:43 4C C5 AB 25 50 55 D4<51>
3430:98 25 47 45 D4 9A 00 82<F2>
3438:80 AC A4 BA BB 9B 47 4F<09>
3440:54 CF 47 4F 53 55 C2 54<19>
3448:CF 53 54 45 D0 54 48 45<6F>
3450:CE A3 3C BD 3C BE 3E BD<AD>
3458:BC BE BD DE AA AB AD AF<B7>
3460:4E 4F D4 4F D2 41 4E C4<F7>
3468:A8 A9 BD BD 3C BD 3C BE<41>
3470:3E BD BC BE BD AB AD A8<30>
3478:80 80 A8 A8 AC 53 54 52<BD>
3480:A4 43 48 52 A4 55 53 D2<6A>
3488:41 53 C3 56 41 CC 4C 45<CA>
3490:CE 41 44 D2 41 54 CE 43<0B>
3498:4F D3 50 45 45 CB 53 49<E1>
34A0:CE 52 4E C4 46 52 C5 45<0F>
34A8:58 D0 4C 4F C7 43 4C 4F<D7>
34B0:C7 53 51 D2 53 47 CE 41<64>
34B8:42 D3 49 4E D4 50 41 44<D6>
34C0:44 4C C5 53 54 49 43 CB<44>
34C8:50 54 52 49 C7 53 54 52<C8>
34D0:49 C7 44 50 45 45 CB A6<96>
34D8:A1 49 4E 53 54 D2 49 4E<9E>
34E0:4B 45 59 A4 45 58 4F D2<7C>
34E8:48 45 58 A4 44 45 C3 44<DD>
34F0:49 D6 46 52 41 C3 54 49<55>
34F8:4D 45 A4 54 49 4D C5 4D<D3>
3500:4F C4 45 58 45 C3 52 4E<C6>
3508:C4 52 41 4E C4 54 52 55<5D>
3510:4E C3 25 B0 25 B1 25 B2<59>
3518:25 B3 47 4F A3 55 49 4E<EA>
3520:53 54 D2 45 52 D2 45 52<45>
3528:CC 00 2B 8C 2C 8B 02 89<2B>
3530:8C 02 8A 8B 03 25 0F 35<67>
3538:02 26 0F 36 02 28 03 B0<C9>
3540:02 8D 02 81 02 B4 03 88<83>
3548:8C 02 03 82 8E 03 2B 0F<D7>
3550:38 8C 8F 2C 02 03 12 0F<DD>
3558:3C 8C 02 03 58 BA 02 6B<4C>
3560:BA 02 60 02 66 02 67 02<5E>
3568:68 02 69 02 6C 02 6D 02<9A>
3570:B1 02 5E B2 02 64 B2 02<9F>
3578:65 B2 02 44 B2 02 D1 B3<F0>
3580:02 48 0F 63 03 3F D0 D3<8D>
3588:2C 03 D0 8C 2C 03 D0 B5<CE>
3590:2C 03 B5 BB B5 03 59 02<F9>
3598:5F 02 B6 02 B7 02 83 03<D8>
35A0:D2 B2 03 85 B9 03 2B 0F<B7>
35A8:37 8C 8F 2C 02 03 1D 0F<8A>
35B0:2F 02 1E 0F 30 02 1F 0F<B5>
35B8:31 02 20 0F 32 02 21 0F<0A>
35C0:33 02 22 0F 34 03 C0 C6<6C>
35C8:90 03 8C 90 03 8D 22 0F<45>
35D0:2D 8C 90 02 B7 22 0F 2E<8C>
35D8:B5 90 03 82 22 0F 2D 8C<3C>
35E0:19 8C BC 90 03 1A 8C 02<14>
35E8:03 C4 12 97 03 C4 12 AF<6E>
35F0:90 03 BE D8 82 90 03 90<24>
35F8:02 91 02 1C B8 03 D9 BF<CC>
3600:D6 90 03 14 02 16 03 BE<49>
3608:90 02 C0 CB 90 03 1C 8C<B4>
3610:03 82 02 85 03 BF D6 02<A7>
3618:03 12 D5 02 03 8C 12 BE<71>
3620:12 C4 12 B5 90 03 BE 90<A8>
3628:02 90 03 96 02 90 03 BE<85>
3630:D8 02 03 BE D8 02 83 D8<EB>
3638:02 03 C1 91 02 C1 92 02<E1>
3640:C1 8C 12 90 02 96 02 90<75>
3648:03 B5 12 02 03 BE 12 82<7A>
3650:12 97 03 8C 12 8C 03 90<1D>
3658:02 8C 12 8C 12 C4 90 03<A9>
3660:C8 D7 90 03 8C C5 90 03<44>
3668:86 C3 03 12 C2 02 03 17<04>
3670:C6 02 18 C6 02 62 C2 02<8A>
3678:6A C2 03 8C C7 03 12 C6<4F>
3680:02 03 82 2B 0F 39 8C 8F<A5>
3688:2C 02 85 2B 0F 3B 8C 2C<16>
3690:03 C8 D7 02 03 12 C9 02<5F>
3698:03 8C 1B CA 90 02 8C 90<DC>
36A0:03 81 02 84 AE 02 CD AD<22>
36A8:02 03 AE 02 03 AF CC 03<19>
36B0:8C 02 B5 03 CD AD 02 03<ED>
36B8:D8 CE 03 CD 02 03 12 02<F2>
36C0:15 03 40 02 41 02 42 02<2E>
36C8:43 02 5C 03 3D 02 3E 02<60>
36D0:5B 03 8C D4 03 12 0F 3C<33>
36D8:D3 02 03 2B 0F 3A 03 BE<56>
36E0:12 92 03 80 D0 B5 12 0F<DA>
36E8:3C B5 8F 2C 03 87 86 90<07>
36F0:03 26 90 02 25 90 02 90<95>
36F8:03 8C 12 93 02 93 03 11<A9>
3700:E9 8F EA 72 E9 D7 EA 04<65>
3708:E9 76 E9 6E E9 11 EB F2<9F>
3710:F0 6E EE 78 EE 80 EE 63<B3>
3718:EE 84 EE 87 EE 8F EE 3C<CA>
3720:EF 03 EF 96 EF 94 EF 90<45>
3728:EF 1C F0 5C EF 2D EF 37<19>
3730:EF 38 EF 06 EF D2 EF 14<12>
3738:EF 73 EF 99 EF 5F EF 58<8D>
3740:EF 56 EF 2B EF 86 EF 9D<54>
3748:EF 40 EF 44 EF 26 F0 30<74>
3750:EF 64 EF 2A F0 FF EE 22<08>
3758:EF E3 EF E6 EF E9 EF 95<C1>
3760:EE BE EE C3 EE C7 EE CB<A0>
3768:EE CF EE D9 EE DC EE 27<D2>
3770:F0 DF EE 1D F0 E7 EE 1E<33>
3778:EF 32 EF 47 EF 4A EF 68<23>
3780:EF 82 EF A1 EF A4 EF 8C<2E>
3788:EF A8 EF B4 EF B7 EF BB<60>
3790:EF CA EF DA EF DD EF ED<1A>
3798:EF F1 EF F4 EF 18 F0 14<23>
37A0:F0 FB EF 05 F0 0B F0 0E<8C>
37A8:F0 4E EF 52 EF CE EF F7<3C>
37B0:EF 6C EF 7E EF 26 EF 32<86>
37B8:F0 20 2B E9 A5 B0 C9 1C<B5>
37C0:90 1C C9 28 F0 18 C9 2B<B5>
37C8:90 16 C9 56 F0 12 C9 57<04>
37D0:F0 0E C9 5A F0 0A C9 5D<14>
37D8:F0 06 C9 61 F0 02 38 60<46>
37E0:4C 60 E9 20 1E F4 A0 00<6D>
37E8:84 A0 84 A1 84 B9 88 84<44>
37F0:AD A9 7F 85 AE 8D FE 02<98>
37F8:20 94 C2 A4 A8 C8 C4 A7<6A>
3800:B0 2E A5 A8 48 20 00 E4<89>
3808:68 85 A8 A5 D2 10 06 20<97>
3810:BE C2 4C 35 F1 20 FB E4<0F>
```

```
3818:84 A1 85 A0 A6 A8 E4 A7<08>
3820:F0 0A A0 80 E8 E4 A7 F0<23>
3828:03 20 FB E4 85 AD 84 AE<5D>
3830:20 62 C9 A0 01 B1 8A 30<3E>
3838:34 C5 AE 90 0B D0 2E 88<5E>
3840:B1 8A C5 AD 90 02 D0 25<68>
3848:AD 1F D0 C9 07 F0 06 A5<16>
3850:14 29 0F D0 F3 20 07 F2<08>
3858:A5 11 F0 0F A0 02 B1 8A<3E>
3860:18 65 8A 85 8A 90 CC E6<CD>
3868:8B B0 C8 C6 11 A5 B5 F0<E5>
3870:07 20 F8 C4 A9 00 85 B5<AC>
3878:8D FE 02 20 D3 F7 4C 2B<1E>
3880:F8 86 AA 85 96 84 95 A4<69>
3888:AA A5 AF F0 10 C6 AF B1<B6>
3890:95 30 03 C8 D0 F9 C8 20<C7>
3898:D7 F1 4C C1 F1 18 98 65<41>
38A0:95 85 95 90 02 E6 96 60<FA>
38A8:A0 FF 84 AF E6 AF A4 AF<83>
38B0:B1 95 48 C9 9B F0 04 29<59>
38B8:7F F0 03 20 96 C2 68 10<D8>
38C0:EB 60 20 02 F2 20 E2 F1<86>
38C8:A9 20 4C 96 C2 A0 00 B1<B5>
38D0:8A AA C8 B1 8A 20 52 F9<1E>
38D8:AE 91 F3 F0 11 A6 B9 A0<D5>
38E0:04 B1 8A 20 9A F3 F0 04<66>
38E8:C9 40 D0 02 CA CA 86 D4<FD>
38F0:20 02 F2 C6 D4 10 F9 A0<FC>
38F8:02 B1 8A 85 9F C8 B1 8A<22>
3900:85 A7 C8 84 A8 20 49 F2<E1>
3908:A4 A7 C4 9F 90 F0 60 20<BD>
3910:3A F3 C9 36 F0 32 C9 54<06>
3918:F0 17 20 46 F3 20 3A F3<20>
3920:C9 37 F0 04 C9 02 B0 20<10>
3928:20 38 F3 20 96 C2 4C 62<CF>
3930:F2 A0 1E AD 91 F3 D0 02<0C>
3938:A0 02 84 D4 A9 2D 20 96<01>
3940:C2 C6 D4 D0 F7 4C 94 C2<6E>
3948:20 38 F3 D0 04 20 38 F3<38>
3950:2C 10 0F 49 80 20 B1 F3<1C>
3958:C9 A8 D0 EC 20 38 F3 4C<E5>
3960:82 F2 C9 0F F0 2E B0 55<08>
3968:48 20 60 E4 C6 A8 68 C9<7D>
3970:0D D0 12 A9 24 20 96 C2<B8>
3978:20 25 33 20 05 DA 09 80<90>
3980:99 7F 05 D0 03 20 BE 2B<BB>
3988:20 5C F9 4C 82 F2 20 E2<7B>
3990:F1 4C 82 F2 20 38 F3 85<8A>
3998:AF A9 22 20 96 C2 A5 AF<08>
39A0:F0 13 20 38 F3 C9 22 D0<25>
39A8:05 20 96 C2 A9 22 20 96<37>
39B0:C2 C6 AF D0 ED A9 22 20<F5>
39B8:96 C2 4C 82 F2 38 E9 10<D9>
39C0:85 AF A2 00 A9 ED A0 71<9E>
39C8:20 BB F1 20 3A F3 C9 5A<E2>
39D0:F0 23 C9 5D F0 1F C9 61<B0>
39D8:F0 1B C9 62 F0 17 C9 6A<25>
39E0:F0 13 C9 28 F0 12 C9 3D<41>
39E8:B0 A4 A0 00 B1 95 29 7F<BC>
39F0:20 57 EA B0 99 20 02 F2<83>
39F8:20 FF F1 4C 82 F2 E6 A8<64>
3A00:A4 A8 C4 A7 B0 03 B1 8A<C8>
3A08:60 68 68 60 48 20 59 F3<7F>
3A10:68 85 AF A2 01 A9 EB A0<EA>
3A18:5A 20 BB F1 4C FF F1 C9<11>
3A20:08 F0 1F C9 3C F0 1B C9<A5>
3A28:3E F0 17 C9 45 F0 13 C9<F8>
3A30:4F F0 0F C9 07 F0 12 20<F2>
3A38:9A F3 D0 0C C6 B9 C6 B9<CF>
3A40:10 06 E6 B9 E6 B9 30 F4<EC>
3A48:60 A5 A8 48 20 66 35 68<F4>
3A50:85 A8 E0 1B D0 EC 60 01<CE>
3A58:B1 8A 49 26 8D 91 F3 60<07>
3A60:C9 09 F0 12 C9 3D F0 0E<C3>
3A68:C9 3F F0 0A C9 46 F0 06<F4>
3A70:C9 51 F0 02 C9 41 60 85<4C>
3A78:AF A2 00 A5 83 A4 82 20<FC>
3A80:BB F1 4C E2 F1 84 B3 B1<E6>
3A88:8A D0 03 C8 B1 8A 49 80<95>
3A90:85 C7 20 00 E4 A6 90 86<19>
3A98:C4 A6 91 86 C5 20 D3 F7<85>
3AA0:B0 0C D0 0A A0 0C B1 90<30>
3AA8:C5 C7 D0 F1 F0 08 A5 C4<66>
3AB0:85 90 A5 C5 85 91 A9 0D<9A>
3AB8:20 54 F6 20 6B E5 A0 00<F1>
3AC0:20 80 F6 20 BB 2F A6 A8<AB>
3AC8:E8 E4 A7 B0 03 20 6B E5<6B>
3AD0:A0 06 20 80 F6 AD 07 F7<FB>
3AD8:D0 2C A5 C7 A0 0C 91 C4<5A>
3AE0:A9 00 F0 05 4A A4 A8 84<A7>
3AE8:B3 48 A9 04 20 54 F6 68<A3>
3AF0:A0 00 91 C4 A5 8A C8 91<F6>
3AF8:C4 A5 8B C8 91 C4 A6 B3<17>
3B00:CA 8A C8 91 C4 60 A5 90<80>
3B08:48 A5 91 48 A5 C4 85 90<C9>
3B10:A5 C5 85 91 A5 D4 48 A5<EE>
3B18:C7 20 B3 E4 68 20 E5 F4<98>
3B20:90 08 68 85 91 68 85 90<F3>
3B28:B0 B0 68 68 A9 08 A2 09<18>
3B30:20 08 F7 B1 8A D0 03 C8<04>
3B38:B1 8A 49 80 45 C7 F0 C5<C5>
3B40:4C 5A F7 A9 0D 4C DC F8<36>
3B48:A5 BE 85 8A A5 BF 85 8B<98>
3B50:60 B1 8A D0 03 C8 B1 8A<94>
3B58:49 80 85 C7 20 D3 F7 B0<3A>
3B60:E2 D0 E0 A0 0C B1 90 C5<F9>
3B68:C7 D0 F1 A0 06 B1 90 48<F8>
3B70:85 E0 C8 B1 90 85 E1 C8<61>
3B78:B1 90 85 E2 C8 B1 90 85<E3>
3B80:E3 C8 B1 90 85 E4 C8 B1<75>
3B88:90 85 E5 A5 C7 20 B3 E4<1F>
3B90:20 9D 2C B0 13 20 C0 E5<6B>
3B98:68 20 E5 F4 90 40 A9 11<05>
3BA0:20 24 F5 A9 08 4C 2B F8<16>
3BA8:4C 2E E5 85 EC A0 00 B1<47>
3BB0:90 C5 D4 D0 2A C8 B1 90<74>
3BB8:C5 D5 D0 1C C8 B1 90 C5<80>
3BC0:D6 D0 15 C8 B1 90 C5 D7<5C>
3BC8:D0 0E C8 B1 90 C5 D8 D0<CC>
3BD0:07 C8 B1 90 C5 D9 F0 06<8A>
3BD8:6A 45 EC 45 D4 0A 60 05<B2>
3BE0:D4 45 EC 10 F9 6A 49 80<86>
3BE8:0A 60 18 65 90 85 90 85<FB>
3BF0:0E 90 04 E6 91 E6 0F 60<D2>
3BF8:C8 C4 A7 B0 03 20 DE C2<49>
3C00:A5 88 85 8A A5 89 85 8B<11>
3C08:A9 00 8D 07 F7 8D A3 F5<7B>
3C10:A0 03 84 A7 88 B1 8A 85<0F>
3C18:9F 88 B1 8A 30 15 20 E7<0E>
3C20:F6 20 AF F6 20 9E F6 A9<DD>
3C28:00 85 B7 85 B8 85 B6 60<A4>
3C30:20 C4 F5 4C 6B E6 20 00<3A>
3C38:E4 A6 AA BD 60 21 F0 0A<86>
3C40:A6 A8 E8 E4 A7 B0 19 4C<6B>
3C48:9D FF A4 A8 88 B1 8A C9<AA>
3C50:1B F0 09 A9 07 A2 41 A0<EF>
3C58:40 4C 0A F7 A5 9F 85 A7<C6>
3C60:60 B1 8A 49 26 8D A3 F5<5A>
3C68:60 00 A9 80 85 11 AC A3<12>
3C70:F5 F0 03 4C DC F8 20 C4<37>
3C78:F5 20 94 C2 A9 D8 85 95<4E>
3C80:A9 F5 85 96 20 E2 F1 4C<4A>
3C88:2E F9 A0 01 B1 8A 30 07<74>
3C90:85 BB 88 B1 8A 85 BA A9<32>
3C98:00 85 B4 85 B5 60 53 54<58>
3CA0:4F 50 50 45 44 A0 A0 01<C8>
3CA8:B1 8A 10 EB A5 BA 85 A0<C3>
3CB0:A5 BB 85 A1 20 62 C9 A0<20>
3CB8:02 B1 8A 85 9F 68 68 4C<84>
3CC0:D7 FF 20 3A CA 85 BC 84<6A>
3CC8:BD 60 84 B3 20 6B E5 20<E6>
3CD0:25 33 B0 1C A5 D5 D0 18<B3>
3CD8:A5 D4 F0 14 85 B9 A4 A8<CD>
3CE0:88 B1 8A 48 C6 B9 F0 09<1C>
3CE8:20 68 35 E0 12 F0 F5 68<A1>
3CF0:60 68 C9 62 F0 19 C9 6A<40>
3CF8:F0 1D 48 20 FB E4 68 C9<BB>
3D00:17 F0 05 A9 1E 20 23 F4<68>
3D08:A5 D4 A4 D5 4C A0 FF A9<B4>
3D10:51 20 23 F4 4C 85 F8 4C<8B>
3D18:AF F8 85 A4 18 A5 90 85<50>
3D20:C4 65 A4 A8 A5 91 85 C5<3C>
3D28:69 00 CD E6 02 90 09 D0<8F>
3D30:10 CC E5 02 90 02 D0 09<97>
3D38:85 91 85 0F 84 90 84 0E<92>
3D40:60 A9 02 4C DC F8 A5 D4<F6>
3D48:91 C4 C8 A5 D5 91 C4 C8<C8>
3D50:A5 D6 91 C4 C8 A5 D7 91<F8>
3D58:C4 C8 A5 D8 91 C4 C8 A5<A9>
3D60:D9 91 C4 60 A5 8C 85 8E<1C>
3D68:85 90 85 0E A5 8D 85 8F<92>
3D70:85 91 85 0F 60 A6 86 86<FF>
3D78:F5 A4 87 84 F6 A6 F5 E4<83>
3D80:88 A5 F6 E5 89 B0 23 A0<E0>
3D88:00 B1 F5 29 FC 91 F5 A0<4B>
3D90:02 A2 06 A9 00 91 F5 C8<03>
3D98:CA D0 FA A5 F5 18 69 08<D7>
3DA0:85 F5 A5 F6 69 00 85 F6<50>
3DA8:D0 D3 4C C5 C9 A0 00 84<FD>
3DB0:BA 84 BB 84 B9 84 FB 84<29>
3DB8:B6 84 B7 84 B8 88 84 BD<ED>
3DC0:84 11 4C 34 C5 B1 8A 49<94>
3DC8:26 8D 07 F7 60 00 A0 FF<4C>
3DD0:8E 30 F7 8D 34 F7 8C 38<93>
3DD8:F7 A9 00 85 D7 A5 8A 85<69>
3DE0:BE A5 8B 85 BF A4 A7 C4<C3>
3DE8:9F B0 3D B1 8A 85 A7 C8<B1>
3DF0:B1 8A C8 84 A8 C9 00 F0<07>
3DF8:0D C9 00 F0 12 C9 00 D0<AD>
3E00:E4 A5 D7 D0 E0 60 A5 D7<20>
3E08:F0 FB C6 D7 4C 1F F7 C9<77>
3E10:07 D0 09 A4 A7 88 B1 8A<B2>
3E18:C9 1B F0 C9 E6 D7 D0 C5<99>
3E20:20 82 F4 A9 16 4C DC F8<A7>
3E28:A0 01 B1 8A 30 F2 18 A5<02>
3E30:9F 65 8A 85 8A 90 02 E6<93>
3E38:8B A0 01 B1 8A 30 E1 C8<34>
3E40:B1 8A 85 9F C8 84 A7 4C<1D>
3E48:1F F7 20 1E F4 20 00 E4<F8>
3E50:A6 AA BD 60 21 D0 27 20<58>
3E58:D3 F7 A9 3E A2 3F 4C 08<9F>
3E60:F7 20 00 E4 20 D3 F7 B0<98>
3E68:28 C9 3C D0 B6 A6 AA BC<15>
3E70:60 21 D0 0A A9 04 20 24<59>
3E78:F5 A9 3C 4C 2B F8 60 20<DF>
3E80:D3 F7 B0 10 C9 3E D0 9B<B5>
3E88:20 2B F8 A4 B2 88 84 A7<7D>
3E90:60 A9 17 2C A9 18 4C DC<24>
3E98:F8 A5 8F C5 91 90 06 A5<0F>
3EA0:8E C5 90 B0 37 38 A5 90<F3>
3EA8:E9 04 85 90 85 0E B0 04<EA>
3EB0:C6 91 C6 0F A0 03 B1 90<19>
3EB8:85 B2 88 B1 90 85 A1 88<ED>
3EC0:B1 90 85 A0 88 B1 90 D0<3B>
3EC8:11 A8 38 A5 90 E9 0D 85<0F>
3ED0:90 85 0E B0 04 C6 91 C6<50>
3ED8:0F 98 18 60 38 60 20 D3<21>
3EE0:F7 B0 2D C9 0C F0 0A C9<F3>
3EE8:1E F0 06 C9 50 D0 EF F0<5C>
3EF0:1F A4 B2 D1 A0 D0 16 A5<5D>
3EF8:A1 F0 12 85 8B A5 A0 85<10>
3F00:8A 88 B1 8A 85 A7 A0 02<38>
3F08:B1 8A 85 9F 60 A9 0F 2C<5C>
3F10:A9 10 2C A9 1A 4C DC F8<21>
3F18:20 D3 F7 B0 F6 D0 02 A9<D1>
3F20:08 AA E8 4C 08 F7 A9 1E<5A>
3F28:D0 C7 20 D3 F7 B0 14 C9<DB>
3F30:50 F0 BE C9 51 F0 EF C9<00>
3F38:1E F0 08 C9 0C D0 EB 38<0C>
3F40:E9 A4 2C A9 1C 4C DC F8<DF>
3F48:20 1E F4 A2 C1 A4 A8 B1<8F>
3F50:8A D0 03 C8 B1 8A 49 80<22>
3F58:20 ED E5 8A D1 9D D0 DF<EF>
3F60:A0 03 B1 9D 85 8B 88 B1<6D>
3F68:9D 85 8A B1 8A 85 9F C8<FE>
3F70:B1 8A 85 A7 60 A2 C2 D0<DE>
3F78:D4 A9 1B 2C A9 19 4C DC<47>
3F80:F8 20 D3 F7 B0 F6 C9 45<DA>
3F88:D0 0A A9 04 20 24 F5 A9<6A>
3F90:45 4C 2B F8 4C 5D F7 A9<D3>
3F98:15 2C A9 13 2C A9 12 2C<44>
3FA0:A9 11 85 B9 A9 00 D8 8D<D5>
3FA8:FE 02 20 C4 F5 A4 BD 30<C7>
3FB0:11 A5 BC A2 80 86 BD A6<AF>
3FB8:B9 86 C3 A2 00 86 B9 4C<7A>
3FC0:A0 FF A5 B9 C9 80 D0 03<EA>
3FC8:4C B0 F5 20 94 C2 A9 37<2E>
3FD0:20 46 F3 A6 B9 A9 00 20<39>
3FD8:52 F9 A5 B9 C9 1F 90 02<D3>
3FE0:E9 62 85 AF C9 4C B0 0C<EB>
3FE8:A2 00 A9 F9 A0 67 20 BB<71>
3FF0:F1 20 FC F1 A0 01 B1 8A<6B>
3FF8:30 15 A9 67 A0 F9 20 60<5F>
4000:F9 A0 01 B1 8A 85 D5 88<2C>
4008:B1 8A 85 D4 20 56 F9 20<1B>
4010:94 C2 20 66 FF 4C 7B E6<AB>
4018:85 D5 86 D4 20 DC 2B 20<75>
4020:BE 2B A5 F3 A4 F4 85 95<F9>
4028:84 96 4C E2 F1 20 41 54<D4>
4030:20 4C 49 4E 45 A0 3F 42<3A>
4038:4C 4F 41 C4 4D 45 CD 56<7E>
4040:41 4C 55 C5 3E 23 56 41<D5>
4048:52 D3 24 4C 45 CE 3F 44<0B>
4050:41 54 C1 3E 33 32 37 36<55>
4058:B7 49 4E 50 55 D4 44 49<E0>
4060:CD 53 54 41 43 CB 4F 56<D6>
4068:45 52 46 4C 4F D7 3F 4C<AD>
4070:49 4E C5 3F 46 4F D2 54<98>
4078:4F 4F 20 4C 4F 4E C7 3F<3A>
4080:44 45 CC 3F 47 4F 53 55<3F>
4088:C2 47 41 52 42 41 47 C5<D1>
4090:3F 43 48 D2 4D 45 CD A3<1A>
4098:3F 4C 4F 41 C4 3F 4E 45<8B>
40A0:53 D4 3F 57 48 49 4C C5<FF>
40A8:3F 52 45 50 45 41 D4 3F<FF>
40B0:44 CF 3F 45 58 49 D4 58<1B>
40B8:50 52 4F C3 3F 45 58 45<0F>
40C0:C3 3F 50 52 4F C3 3F A3<37>
```

Listing »Turbo-Basic XL« (Fortsetzung)

```
40C8:49 53 20 4F 50 45 CE 3F<EE>
40D0:44 45 D6 57 52 20 4F 4E<C3>
40D8:4C D9 43 4D C4 4E 4F 54<2B>
40E0:20 4F 50 45 CE A3 52 44<37>
40E8:20 4F 4E 4C D9 45 4F C6<DB>
40F0:54 52 55 4E C3 54 49 4D<83>
40F8:45 4F 55 D4 4E 41 CB 21<7C>
4100:46 52 41 4D C5 43 55 52<46>
4108:53 4F D2 21 4F 56 45 52<C2>
4110:52 55 CE 43 48 4B 53 55<21>
4118:CD 44 4F 4E C5 4D 4F 44<12>
4120:C5 4E 4F 54 20 49 4D 50<AE>
4128:CC 52 41 CD A0 A0 A0 A0<7C>
4130:A0 A0 A0 A0 A0 A0 A0 A0<1D>
4138:44 3F BA 3E 23 46 49 4C<CE>
4140:45 D3 44 53 4B 20 46 55<6C>
4148:4C CC 46 41 54 41 4C 20<0C>
4150:49 2F CF 46 49 4C 45 A3<F6>
4158:4E 41 4D C5 50 4F 49 4E<5E>
4160:D4 4C 4F 43 4B 45 C4 44<10>
4168:43 4D C4 3E 44 49 D2 3F<E5>
4170:46 49 4C C5 50 4F 49 4E<64>
4178:D4 3F 41 50 50 4E C4 42<57>
4180:41 44 20 53 45 43 54 4F<17>
4188:52 D3 A5 14 A4 13 A6 12<A4>
4190:C5 14 D0 F6 48 86 D5 84<DD>
4198:D4 20 DC 2B A2 05 BD 68<44>
41A0:FB 95 E0 CA 10 F8 20 9E<AA>
41A8:26 20 9C 2E 68 85 D4 A9<21>
41B0:00 85 D5 20 DC 2B 20 9D<58>
41B8:2C A9 00 85 D2 85 D3 4C<99>
41C0:3A E5 20 C4 FA C6 AA A2<B9>
41C8:05 BD 6E FB 95 E0 CA 10<69>
41D0:F8 20 00 28 A0 00 A5 D4<FB>
41D8:C9 40 D0 02 84 D5 84 F2<1B>
41E0:A9 24 20 30 FB A9 60 20<01>
41E8:30 FB A9 60 20 30 FB A0<53>
41F0:06 A9 80 4C 6F DA 48 20<D7>
41F8:B8 2D 20 9C 2E 20 BB 2F<FD>
4200:68 85 D5 20 9E 26 A9 00<4D>
4208:A4 D4 C0 40 D0 05 A4 D5<80>
4210:85 D5 98 AA 4A 4A 4A 4A<92>
4218:20 58 FB 8A 29 0F 09 30<4E>
4220:C9 3A 90 02 69 06 A4 F2<46>
4228:99 80 05 E6 F2 60 41 02<F1>
4230:56 00 00 00 43 04 32 00<66>
4238:00 00 20 00 E4 20 73 C5<F0>
4240:A0 00 84 F2 84 E1 84 E2<B0>
4248:20 C4 FB C9 18 B0 35 85<28>
4250:E0 20 13 FC 20 C4 FB C9<BD>
4258:3C B0 29 20 DE FB 20 13<3C>
4260:FC 20 C4 FB C9 3C B0 1C<C7>
4268:20 DE FB 20 F2 FB 20 EC<7D>
4270:FB A5 E0 A4 E1 A6 E2 85<BC>
4278:14 84 13 86 12 C5 14 D0<FE>
4280:F6 4C 93 C5 20 93 C5 4C<13>
4288:D7 F8 20 D2 2B E6 F2 B0<95>
4290:F3 0A 85 E3 0A 0A 65 E3<03>
4298:85 E3 20 D2 2B E6 F2 B0<3D>
42A0:E3 65 E3 60 18 65 E0 85<DC>
42A8:E0 90 06 E6 E1 D0 02 E6<5F>
42B0:E2 60 06 E0 26 E1 26 E2<DE>
42B8:A4 E2 A5 E0 A6 E1 06 E0<DD>
42C0:26 E1 26 E2 06 E0 26 E1<4F>
42C8:26 E2 65 E0 85 E0 8A 65<88>
42D0:E1 85 E1 98 65 E2 85 E2<6B>
42D8:60 20 EC FB A4 E2 A5 E0<17>
42E0:A6 E1 06 E0 26 E1 26 E2<21>
42E8:65 E0 85 E0 8A 65 E1 85<4D>
42F0:E1 98 65 E2 06 E0 26 E1<72>
42F8:2A 85 E2 60 60 20 0C E5<21>
4300:85 99 D0 F8 20 00 E4 A6<99>
4308:AA BD 00 21 30 03 20 34<26>
4310:DA 20 DC E4 A5 D6 05 D7<A2>
4318:F0 E2 A0 00 84 DB 84 DC<77>
4320:B1 D4 10 02 C6 DB 20 D0<40>
4328:FC 0A 0A 85 A2 A9 00 2A<AA>
4330:06 A2 2A 6D F4 02 85 A3<91>
4338:20 1A FD B0 BF A0 08 84<08>
4340:DD A4 DC B1 A2 45 DB 85<01>
4348:DA A6 ED A4 E3 06 DA B1<A2>
4350:DE 3D C0 05 90 03 1D C8<E6>
4358:05 91 DE C6 DD F0 07 20<82>
4360:0B FF C4 E1 90 E7 20 E0<F3>
4368:FE E6 DC A5 DC C9 08 B0<03>
4370:06 65 99 C5 E0 90 C6 A5<E4>
4378:9B 69 07 85 9B 90 02 E6<E0>
4380:9C E6 D4 D0 02 E6 D5 A5<3B>
4388:D6 D0 02 C6 D7 C6 D6 4C<D3>
4390:4E FC 40 20 60 00 A8 0A<7A>
4398:0A 2A 2A 29 03 AA 98 5D<D2>
43A0:CC FC 60 18 18 0C 18 30<E9>
43A8:30 60 60 C0 C0 C0 C0 18<DE>
43B0:0C C0 C0 28 14 14 0A 0A<B5>
43B8:14 14 28 28 28 28 28 28<2F>
43C0:28 14 28 00 00 00 02 03<4B>
43C8:02 03 02 03 01 01 01 00<8E>
43D0:00 03 02 00 01 03 07 00<79>
43D8:F0 FC FE 04 02 01 38 60<39>
43E0:A5 57 29 0F AA A9 00 85<D9>
43E8:DF A5 99 0A 26 DF 0A 26<C5>
43F0:DF 65 99 90 02 E6 DF 0A<CA>
43F8:85 DE 26 DF BD DD FC 85<01>
4400:E0 C5 99 F0 D9 90 D7 BC<CB>
4408:FD FC 84 E2 B9 11 FD 8D<4D>
4410:C0 05 BD ED FC 85 E1 4A<A2>
4418:4A 4A 4A AA F0 07 06 DE<B1>
4420:26 DF CA D0 F9 18 A5 DE<F9>
4428:65 58 85 DE A5 DF 65 59<AC>
4430:85 DF A5 9C 85 E4 A5 9B<21>
4438:85 E3 A4 E2 84 ED F0 0C<5A>
4440:39 0D FD 85 ED 46 E4 66<42>
4448:E3 88 D0 F9 A5 E4 D0 8E<F0>
4450:A5 E3 C5 E1 B0 88 A6 E2<79>
4458:D0 0A A5 C8 20 D0 FC 8D<41>
4460:C8 05 18 60 BC 0D FD 84<A7>
4468:E4 BD 14 FD 85 EE A5 C8<53>
4470:0D C0 05 4D C0 05 99 C8<F0>
4478:05 A6 EE 0A CA D0 FC 88<98>
4480:10 F4 A4 E4 AD C0 05 99<D7>
4488:C0 05 A6 EE 38 2A CA D0<13>
4490:FB 88 10 F3 18 60 20 0C<77>
4498:E5 85 99 D0 F8 20 1A FD<71>
44A0:B0 F3 A5 90 85 A2 A5 91<27>
44A8:85 A3 AD E5 02 E9 06 85<D4>
44B0:E7 AD E6 02 E9 00 85 E8<EC>
44B8:18 A5 A2 69 03 85 A2 90<73>
44C0:02 E6 A3 C5 E7 A5 A3 E5<77>
44C8:E8 90 03 4C 7B F6 A6 ED<71>
44D0:A4 E3 20 26 FF F0 03 4C<53>
44D8:BA FE 20 1B FF 20 14 FF<F1>
44E0:98 30 0B 20 26 FF D0 06<36>
44E8:20 1B FF 4C 17 FE 20 0B<67>
44F0:FF 98 A0 00 91 A2 8A 0A<03>
44F8:0A 0A C8 91 A2 A4 E3 A6<29>
4500:ED 20 0B FF C4 E1 B0 0B<6A>
4508:20 26 FF D0 06 20 1B FF<EE>
4510:4C 3B FE 20 14 FF 98 A0<AD>
4518:02 91 A2 8A 88 11 A2 91<09>
4520:A2 A4 99 C8 C4 E0 B0 23<B6>
4528:20 DE FE 20 FA FE A0 01<46>
4530:B1 A2 29 07 C5 ED C8 B1<53>
4538:A2 E5 E3 90 0B A0 00 B1<A9>
4540:A2 09 80 91 A2 4C F2 FD<F1>
4548:20 EC FE A4 99 88 C4 E0<26>
4550:B0 20 20 EC FE 20 FA FE<DF>
4558:A0 01 B1 A2 29 07 C5 ED<FF>
4560:C8 B1 A2 E5 E3 90 08 A0<C7>
4568:00 B1 A2 29 7F 10 D4 20<E3>
4570:DE FE A0 01 B1 A2 29 07<F2>
4578:AA C8 B1 A2 A8 20 0B FF<74>
4580:20 0B FF 86 ED 84 E3 38<8C>
4588:A5 A2 E9 03 85 A2 B0 02<AE>
4590:C6 A3 C5 90 D0 06 A5 A3<5C>
4598:C5 91 F0 15 A0 00 B1 A2<40>
45A0:10 B6 30 8A E6 99 18 A5<98>
45A8:DE 65 E1 85 DE 90 02 E6<06>
45B0:DF 60 C6 99 38 A5 DE E5<DA>
45B8:E1 85 DE B0 02 C6 DF 60<CF>
45C0:A0 00 B1 A2 29 7F 85 E3<89>
45C8:C8 B1 A2 4A 4A 4A 85 ED<CE>
45D0:60 E4 E4 E8 90 03 A2 00<EB>
45D8:C8 60 CA 10 03 A6 E4 88<9B>
45E0:60 B1 DE 3D C0 05 1D C8<E8>
45E8:05 91 DE 60 B1 DE 1D C0<D4>
45F0:05 5D C0 05 F0 04 AD C8<0D>
45F8:05 60 AD C8 05 F0 03 A9<D6>
4600:00 60 A9 01 60 20 45 DF<D6>
4608:AD FE 02 48 8E FE 02 20<38>
4610:9C C4 A0 7D 20 9C C2 68<EE>
4618:8D FE 02 4C A9 C2 B1 8A<29>
4620:C9 26 F0 08 A9 4C A2 76<50>
4628:A0 FF D0 06 A9 A0 A2 02<CD>
4630:A0 B1 8D AF FF 8E B0 FF<06>
4638:8C B1 FF 60 A9 5B 20 96<65>
4640:C2 A0 00 B1 8A AA C8 B1<E0>
4648:8A 20 52 F9 A9 5D 20 96<0C>
4650:C2 A0 02 B1 8A 4C B3 FF<DC>
4658:20 82 F4 A9 0C 4C DC F8<AF>
4660:20 1E F4 20 FB E4 85 A0<D7>
4668:84 A1 20 62 C9 B0 E9 68<2F>
4670:68 A5 11 F0 1E A0 02 B1<1F>
4678:8A 85 9F C8 C4 9F B0 1D<34>
4680:B1 8A 85 A7 C8 B1 8A C8<42>
4688:84 A8 20 D0 FF A4 A7 A5<BE>
4690:11 D0 E9 4C A4 F5 0A 8D<14>
4698:D5 FF 6C 00 22 A0 01 B1<22>
46A0:8A 30 12 18 A5 9F 65 8A<2D>
46A8:85 8A 90 02 E6 8B B1 8A<B0>
46B0:10 C3 4C 6A F5 4C 78 E6<05>
Laenge 18104 Bytes
```

Listing »Turbo-Basic-XL« (Schluß)

DR. NIBBLE & crew







ASS – klein, aber fein

Erst durch einen Assembler kommt man mit dem Atari so richtig auf seine Kosten. Wir stellen einen Assembler vor, der zum ersten Reinschnuppern wie geschaffen ist.

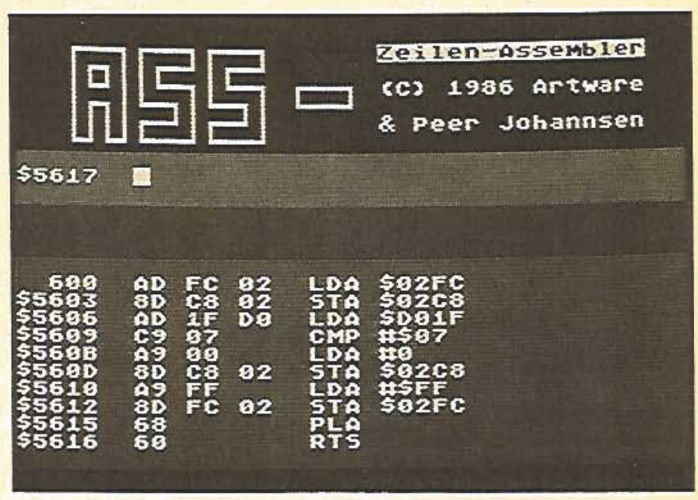
Um einen Computer voll ausnutzen zu können, läßt es sich nicht umgehen, ihn in Maschinensprache zu programmieren. Der Atari-Computer macht hier keine Ausnahme, denn solche Leckerbissen wie Display-List-Interrupts oder Player-Missile-Grafik lassen sich kaum in Basic schreiben. Da es recht schwer ist, in Maschinensprache zu programmieren, benötigt man für das Vorhaben ein Programm, das die unverständlichen Zahlencodes des Mikroprozessors in eine lesbare Form bringt: einen Assembler.

Unser Assembler ist in Turbo-Basic geschrieben. Er eignet sich hervorragend, um kleine Maschinenprogramme zu erzeugen. Zum Beispiel lassen sich damit die in dieser Ausgabe angesprochenen Programmierbeispiele sehr gut ausprobieren. Da es sich um einen Zeilenassembler handelt, ist er für größere Programme weniger geeignet. »Zeilenassembler« bedeutet, daß eine eingegebene Zeile sofort in den Maschinencode des 6502-Prozessors übersetzt wird. Zum Austesten kleiner Programme ist das ideal, da man sich so nicht mit allen möglichen Bedienungsbefehlen anderer Assembler herumschlagen muß.

Der Assembler besteht aus drei Teilen

Der »ASS-Assembler« besteht aus drei Teilprogrammen. Tippen Sie alle drei unter Turbo-Basic und mit Hilfe des Prüfsummen-Programms ab. Das erste Programm mit dem Namen »DATEI.SET« ist lediglich dazu da, eine Hilfsdatei auf der Diskette zu erzeugen. Wenn Sie das Programm laufen lassen, wird auf der Diskette das File »DLI.DAT« erzeugt, das vom Assembler benötigt wird.

Das zweite File mit dem Namen »AUTORUN.BAS« wird von Turbo-Basic automatisch geladen und gestartet. Bei dem Programm handelt es sich um ein Vorprogramm, das verschiedene Speicherstellen im Computer setzt und anschließend das eigentliche Assembler-Programm lädt. Am besten verwenden Sie für das Programm eine eigene Diskette. Darauf kopieren Sie dann ein DOS, das Turbo-Basic, das »AUTORUN.BAS«-File, das »DLI.DAT«-File, das von dem »DATEI.SET«-Programm erzeugt wurde, und den Assembler. Wenn Sie das Turbo-Basic in »AUTORUN.SYS« umbenennen, lädt der Computer beim Einschalten den



Kleine Programme sind mit »ASS« kein Problem

Assembler automatisch. Unter Happy-DOS müssen Sie den Befehl »JOB TURBO.COM« benutzen, damit Turbo-Basic und damit auch der Assembler automatisch geladen und gestartet werden.

Wenn der ASS-Assembler geladen ist, sehen Sie einen viergeteilten Bildschirm. Der oberste braune Teil enthält lediglich den Programmnamen. Darunter befindet sich eine blaue Zeile, in der die einzelnen Befehle eingegeben werden. In der folgenden, grau dargestellten Zeile stehen eventuell auftretende Fehlermeldungen des Assemblers. Im unteren grünen Teil wird schließlich ein eingegebenes Programm gelistet. Zusätzlich dient es als Sichtfeld für den eingebauten Monitor.

Der Assembler kennt sämtliche 6502-Befehle. Zur Verdeutlichung zeigt Tabelle 1 die Schreibweisen aller Adres-

```
$5600  LDA $02FC
$5603  STA $02C8
$5606  LDA $D01F
$5609  CMP #$07
$560B  BEQ $5600
$560D  LDA #$00
$560F  STA $02C8
$5612  LDA #$FF
$5614  STA $02FC
$5617  PLA
$5618  RTS
```

Das Programm wird mit der <START>-Taste beendet

Assembler Syntax und Adressierungsarten	
NOP	– Implied
LDA #$A5	– Immediate
LDX $5600	– Absolut
STA $05	– Zeropage
LDA $0200,X	– Absolut,X
LDA $0200,Y	– Absolut,Y
LDA $10,X	– Zeropage,X
LDA $10,Y	– Zeropage,Y
JMP $0600	– Jump direct
JMP ($0600)	– Jump indirect
ROR A	– Accu
BNE $5600	– Branch

Tabelle 1 (oben), Tabelle 2 (rechts)

Assemblerbefehle	
PON	– Alle Befehle werden zusätzlich auf dem Drucker ausgegeben
POF	– Drucker wird ausgeschaltet
ORG $5800	– Programmzeiger setzen
COD 00101101	– Bitmuster in die Speicherstelle des momentanen Programmzeigers schreiben
DAT $FF,$12, $A1,$2B	– Hexadezimalzahlen in die Speicherstellen des momentanen Programmzeigers schreiben
ASC Dies ist ein Text	– ASCII in die Speicherstellen des momentanen Programmzeigers schreiben
CHA $0600	– Zeigt die entsprechende Dezimalzahl
CHA 1536	– Zeigt die entsprechende Hexadezimalzahl
MOV $1000,100	– Verschiebt einen beliebigen Block (in diesem Beispiel von $1000 ab 100 Byte zur Stelle hin, auf die der Programmzeiger zeigt)
MON	– Aufruf des Monitors
DIR	– Zeigt Directory von Laufwerk 1
DOS	– Geht ins DOS
END	– Beendet das Programm und geht zurück ins Basic
NEW	– Startet den Assembler neu und beginnt wieder bei Adresse $5600
FIL 2	– Schreibt in Speicherstellen hinter dem Programmzeiger zweimal den Wert des Programmzählers
PUT D: PRG.DAT,100	– Speichert einen Speicherinhalt und zwar 100 Byte vom Beginn des Programmzeigers ab unter dem Namen PRG.DAT auf Diskette
GET D: PRG.DAT	– Lädt den Programmblock PRG:DAT von Diskette. Anfangs- und Endadresse werden dem Block entnommen

sierungsarten des Mikroprozessors. Ein Dollar-Zeichen vor einer Zahl gibt dabei an, daß es sich um eine Hexadezimalzahl handelt. Bei den Adressierungsarten »Immediate«, »Absolut«, bei den direkten Sprungbefehlen sowie bei den »Branch«-Anweisungen läßt der Assembler anstelle von Hexadezimalzahlen auch dezimale Zahlen zu.

Kommentare lassen sich ebenfalls in ein Programm einfügen. Dies geschieht, indem man einem Text ein Semikolon voranstellt. Sinnvoll ist dies allerdings nur, wenn man mit dem Befehl »PON« einen angeschlossenen Drucker aktiviert. Dann werden alle eingegebenen Befehle und Kommentare gleichzeitig auf dem Drucker ausgegeben.

Ab $5600 ist alles frei

Neben diesem Befehl hat der ASS-Assembler noch weitere Befehle eingebaut. In Tabelle 2 sind sämtliche Kommandos zusammengestellt.

Wenn Sie mit dem Assembler programmieren wollen, sollten Sie zwei Dinge beachten. Zum einen dürfen Sie nicht die Seite 6 im Computerspeicher verwenden, also den Speicherbereich zwischen $600 und $6FF. Hier liegen einige Maschinensprache-Routinen des Assemblers. Wenn Sie dort etwas hineinschreiben, hängt sich das Programm unweigerlich auf. Sie sollten auch darauf verzichten, den Programmzähler niedriger als Speicherstelle $5600 zu setzen, da in diesem Bereich der Assembler steht. Zum anderen kehrt ein von Ihnen geschriebenes Programm nur dann wieder zum Assembler zurück, wenn es mit dem Maschinensprache-Befehl »RTS« beendet wurde. Nebenbei sollten Sie immer mal wieder Ihr Programm mit dem Befehl »PUT« auf Diskette speichern. Es gibt schließlich nichts Unangenehmeres, als durch eine kleine Störung im Stromnetz ein schweißgeborenes Programm zu verlieren.

Zum Ausprobieren des Assemblers haben wir ein kleines Programm abgedruckt. Was das Programm macht, wird nicht verraten. Viel Spaß beim Assemblern!

(Peer Johannsen/hf)

Steckbrief	
Programm:	ASS-Zeilenassembler
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette

```
0 REM ***▲Datei.Set▲***                      <SB>
1 DATA▲76,43,6,72,138,72,166,205,142,10,
212,189,97,6,141,24,208,169,15,253          <XQ>
2 DATA▲97,6,41,15,141,23,208,232,173,11,
212,201,108,208,2,162,0,134,205,104         <PD>
3 DATA▲170,104,64,173,48,2,133,203,173,4
9,2,133,204,160,6,169,130,145,203           <PQ>
4 DATA▲200,192,28,208,247,160,2,169,240,
145,203,160,3,169,194,145,203,169           <LC>
5 DATA▲3,141,0,2,169,6,141,1,2,169,192,1
41,14,212,169,0,133,205,104,96,16           <PN>
6 DATA▲16,16,16,16,16,128,128,128,4,4,4,
194,194,194,194,194,194,194,194,194         <AM>
7 DATA▲194,194,194                          <FO>
8 OPEN #%1,8,%0,"D:DLI.DAT"                 <PY>
9 FOR I=%1 TO 121:READ X:Z=Z+X:PUT #%1,X
:NEXT I:CLOSE #%1                           <GG>
10 IF Z<>14355 THEN ? "DATA-ERROR▲!!!":E
ND                                          <DA>
11 ? "OK▲-▲ALLES▲KLAR"                      <UN>
```

Listing. »DATEI.SET« erzeugt das File »DLI.DAT«

```
5 REM ***▲ASS▲-▲Zeilenassembler▲***          <OU>
10 GRAPHICS 0:POKE 752,1:POKE 710,128:PO
KE 712,128:POSITION 2,7                     <BL>
15 ? "▲{CTL Q}{CTL R}{CTL R}{CTL E}{CTL
Q}{CTL R}{CTL R}{CTL E}{CTL Q}{CTL R}{CT
L R}{CTL E}▲▲▲▲▲▲Zeilen-Assembler"          <AQ>
20 ? "▲{SHIFT =}{CTL Q}{CTL E}{SHIFT =}{
SHIFT =}{CTL Q}{CTL R}{CTL C}{SHIFT =}{C
TL Q}{CTL R}{CTL C}"                        <HL>
25 ? "▲{SHIFT =}{CTL Z}{CTL C}{SHIFT =}{
SHIFT =}{CTL Z}{CTL R}{CTL E}{SHIFT =}{C
TL Z}{CTL R}{CTL E}▲{CTL Q}{CTL R}{CTL R
}{CTL E}▲(C)▲1986▲Artware"                  <YO>
30 ? "▲{SHIFT =}{CTL Q}{CTL E}{SHIFT =}{
CTL Z}{CTL R}{CTL E}{SHIFT =}{CTL Z}{CTL
 R}{CTL E}{SHIFT =}▲{CTL Z}{CTL R}{CTL R
}{CTL C}"                                   <BK>
35 ? "▲{SHIFT =}{SHIFT =}{SHIFT =}{SHIFT
 =}{CTL Q}{CTL R}{CTL C}{SHIFT =}{CTL Q}
{CTL R}{CTL C}{SHIFT =}▲▲▲▲▲▲&▲Peer▲Joha
nnsen"                                      <VF>
40 ? "▲{CTL Z}{CTL C}{CTL Z}{CTL C}{CTL
Z}{CTL R}{CTL R}{CTL C}{CTL Z}{CTL R}{CT
L R}{CTL C}"                                <HA>
45 OPEN #1,4,0,"D:DLI.DAT":BGET #1,1536,
121:CLOSE #1:RUN "D:ASS.PRG"                <QN>
```

Listing. »AUTORUN.BAS« erzeugt ein Ladebild und startet den Assembler

```
0 REM ***▲ASS▲-▲Zeilenassembler▲***          <OP>
5 DIM G$(22),A$(%2),B$(%2),C$(%2),D$(%2)
,H$(5),P$(%2),R$(5)                         <BB>
10 --                                       <YW>
15 # START:GRAPHICS %0:POKE 752,%1:X=USR
(1536):Z=22016                              <YX>
20 ? "▲{CTL Q}{CTL R}{CTL R}{CTL E}{CTL
Q}{CTL R}{CTL R}{CTL E}{CTL Q}{CTL R}{CT
L R}{CTL E}▲▲▲▲▲▲Zeilen-Assembler"          <AH>
25 ? "▲{SHIFT =}{CTL Q}{CTL E}{SHIFT =}{
SHIFT =}{CTL Q}{CTL R}{CTL C}{SHIFT =}{C
TL Q}{CTL R}{CTL C}"                        <HV>
30 ? "▲{SHIFT =}{CTL Z}{CTL C}{SHIFT =}{
SHIFT =}{CTL Z}{CTL R}{CTL E}{SHIFT =}{C
TL Z}{CTL R}{CTL E}▲{CTL Q}{CTL R}{CTL R
}{CTL E}▲(C)▲1986▲Artware"                  <YF>
35 ? "▲{SHIFT =}{CTL Q}{CTL E}{SHIFT =}{
CTL Z}{CTL R}{CTL E}{SHIFT =}{CTL Z}{CTL
 R}{CTL E}{SHIFT =}▲{CTL Z}{CTL R}{CTL R
}{CTL C}"                                   <BU>
40 ? "▲{SHIFT =}{SHIFT =}{SHIFT =}{SHIFT
 =}{CTL Q}{CTL R}{CTL C}{SHIFT =}{CTL Q}
{CTL R}{CTL C}{SHIFT =}▲▲▲▲▲▲&▲Peer▲Joha
nnsen"                                      <UW>
45 ? "▲{CTL Z}{CTL C}{CTL Z}{CTL C}{CTL
Z}{CTL R}{CTL R}{CTL C}{CTL Z}{CTL R}{CT
L R}{CTL C}"                                <HK>
50 --                                       <ZA>
55 # MAIN:POSITION %0,7:? "$";HEX$(Z):EX
EC INPUT:POSITION 7,10:? "▲▲▲▲▲▲▲▲▲▲▲▲▲▲ <FD>
60 IF G$(%1,%1)="▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲;" THE
N GO# PRINT                                 <PG>
65 H$=G$(%1,%3):A$="▲▲":B$=A$:C$=A$         <AF>
70 --                                       <ZC>
75 IF H$="PON" THEN PR=%1:POSITION 7,10:
? "Drucker▲an":GO# MAIN                     <ND>
80 IF H$="POF" THEN PR=%0:POSITION 7,10:
? "Drucker▲aus":GO# MAIN                    <YC>
85 IF H$="ORG" THEN GO# ORG                 <KT>
90 IF H$="COD" THEN GO# COD                 <KU>
95 IF H$="DAT" THEN GO# DAT                 <QS>
100 IF H$="CHA" THEN GO# CHA                <EK>
105 IF H$="ASC" THEN GO# ASC                <HF>
110 IF H$="MOV" THEN GO# MOV                <LU>
115 IF H$="RUN" THEN GO# RUN                <PZ>
120 IF H$="MON" THEN GO# MON                <YY>
```

```
125 IF H$="FIL" THEN GO# FIL                <CV>
130 IF H$="GET" THEN GO# GET                <KQ>
135 IF H$="PUT" THEN GO# PUT                <WX>
140 IF H$="NEW" THEN GO# START              <CZ>
145 IF H$="DIR" THEN GO# DIR                <JT>
150 IF H$="DOS" THEN DOS                    <EL>
155 IF H$="END" THEN TRAP %1:GRAPHICS %0
:END                                        <JO>
160 --                                      <JO>
165 TRAP 570:A$=G$(5,5):B$=G$(6,6):C$=G$
(8,8):D$=G$(11,11)                          <BS>
170 IF A$="▲" OR A$="A" THEN P$="IP":C=%
0                                           <KJ>
175 IF A$="#" AND B$="$" THEN P$="IM":C=
%2                                          <UV>
180 IF A$="#" AND B$>="0" AND B$<="9" TH
EN P$="IM":C=7                              <LF>
185 IF A$="(" AND B$="$" AND D$="Y" THEN
 P$="IY":C=%2                               <AG>
190 IF A$>="0" AND A$<="9" THEN P$="AB":
C=6                                         <UC>
195 IF A$="(" AND B$="$" AND G$(10,10)="
X" THEN P$="IX":C=%2                        <QF>
200 IF A$="$" AND G$(10,10)="▲" THEN P$=
"AB":C=%1                                   <FW>
205 IF A$="$" AND C$="▲" THEN P$="ZP":C=
%3                                          <UQ>
210 IF A$="$" AND C$="," AND G$(9,9)="X"
 THEN P$="ZX":C=%3                          <YO>
215 IF A$="$" AND C$="," AND G$(9,9)="Y"
 THEN P$="ZY":C=%3                          <CQ>
220 IF A$="$" AND D$="X" THEN P$="AX":C=
%1                                          <SR>
225 IF A$="$" AND D$="Y" THEN P$="AY":C=
%1                                          <VN>
230 IF A$="(" AND D$=")" AND G$(10,10)<>
"X" THEN P$="IN":C=5                        <HR>
235 IF H$(%1,%1)="B" AND H$<>"BIT" AND H
$<>"BRK" THEN P$="▲R":C=4                   <ZO>
240 --                                      <JL>
245 RESTORE :H$(4)=P$:I=-%1:REPEAT          <EV>
250 READ R$:I=I+%1:IF I=256 THEN 570        <LJ>
255 UNTIL R$=H$:A$=HEX$(I):B$="▲▲":C$=B$ <YD>
260 --                                      <JP>
265 POKE Z,I:IF C=%0 THEN X=Z:Z=Z+%1:GO#
 PRINT                                      <XW>
270 ON C GO# AB,IM,ZP,RR,IN,DE,ZE           <HQ>
275 GO# MAIN                                <EX>
280 --                                      <JT>
285 # RUN:IF G$(5,5)<>"$" THEN 570          <BT>
290 X=USR(DEC(G$(6,9))):GO# MAIN            <HX>
295 --                                      <KK>
300 # ASC:X=Z:FOR I=22 TO %1 STEP -%1:IF
 G$(I,I)<>"▲" THEN POP :GOTO 310            <LI>
305 NEXT I                                  <GI>
310 FOR C=5 TO I:POKE Z,ASC(G$(C,C)):Z=Z
+%1:NEXT C:GO# PRINT                        <VZ>
315 --                                      <JV>
320 # COD:X=%0:FOR I=5 TO 12:IF G$(I,I)=
"1" THEN X=X+%2^(7-(I-5))                   <PV>
325 NEXT I:POKE Z,X:X=Z:Z=Z+%1:GO# PRINT <WO>
330 --                                      <JK>
335 # DAT:C=%0:FOR I=5 TO 17 STEP 4:IF G
$(I,I)="$" THEN POKE Z+C,DEC(G$(I+%1,I+%
2)):C=C+%1                                  <NL>
340 NEXT I:X=Z:Z=Z+C:GO# PRINT              <ZT>
345 --                                      <KB>
350 # MOV:IF G$(10,10)<>"," OR G$(5,5)<>
"$" THEN 570                                <OY>
355 IF G$(11,11)="$" THEN G$(11)=STR$(DE
C(G$(12)))                                  <DS>
360 I=DEC(G$(6,9)):X=VAL(G$(11))            <CL>
365 IF I>=Z THEN MOVE I,Z,X                 <QV>
370 IF I<Z THEN -MOVE I,Z,X                 <EA>
375 X=Z:GO# PRINT                           <OB>
380 --                                      <JU>
385 # AB:DPOKE Z+%1,DEC(G$(6,9)):B$=HEX$
(PEEK(Z+%1)):C$=HEX$(PEEK(Z+%2)):X=Z:Z=Z
+%3:GO# PRINT                               <HR>
390 --                                      <JW>
395 # IM:X=DEC(G$(7,8)):POKE Z+%1,X:B$=H
EX$(X):X=Z:Z=Z+%2:GO# PRINT                 <LR>
400 --                                      <JF>
405 # ZP:X=DEC(G$(6,7)):POKE Z+%1,X:B$=H
EX$(X):X=Z:Z=Z+%2:GO# PRINT                 <OZ>
410 --                                      <JH>
415 # RR:X=DEC(G$(6,9)):X=Z-X               <TM>
420 IF ABS(X)>128 THEN POSITION 7,10:? "
Offset▲>▲128":GET X:GO# MAIN                <IC>
425 IF X<%0 THEN X=ABS(X)-%2:GOTO 435       <FB>
430 X=255-(X+%1)                            <FH>
435 POKE Z+%1,X:B$=HEX$(X):X=Z:Z=Z+%2       <SL>
440 --                                      <JN>
445 # PRINT:POSITION %0,13:? "{ESC SHIFT
 DEL}":POSITION %0,22:IF G$(%1,%1)=";" T
HEN ? G$:GOTO 455                           <VA>
450 ? "$";HEX$(X);"▲▲";A$;"▲";B$;"▲";C$;
"▲▲";G$                                     <MS>
455 IF PR=%0 THEN GO# MAIN                  <AX>
460 TRAP 575:IF G$(%1,%1)=";" THEN LPRIN
T G$:GO# MAIN                               <JZ>
465 LPRINT "$";HEX$(X);"▲▲";A$;"▲";B$;"▲
";C$;"▲▲";G$:GO# MAIN                       <ZQ>
470 --                                      <JT>
475 # IN:DPOKE Z+%1,DEC(G$(7,10)):B$=HEX
$(PEEK(Z+%1)):C$=HEX$(PEEK(Z+%2)):X=Z:Z=
Z+%3:GO# PRINT                              <AV>
480 --                                      <JV>
485 DATA▲BRKIP,ORAIX,,,,ORAZP,ASLZP,,PHP
IP                                          <HD>
490 DATA▲ORAIM,ASLIP,,,ORAAB,ASLAB,,BPL▲
R,ORAIY,,,                                  <GP>
495 DATA▲ORAZX,ASLZX,,CLCIP,ORAAY,,,,ORA
AX,ASLAX,,JSRAB,ANDIX,,,BITZP,ANDZP         <DA>
500 DATA▲ROLZP,,PLPIP,ANDIM,ROLIP,,BITAB
,ANDAB,ROLAB,,BMI▲R,ANDIY,,,                <OA>
505 DATA▲ANDZX,ROLZX,,SECIP,ANDAY,,,,AND
AX,ROLAX,,RTIIP,EORIX,,,,EORZP,LSRZP,       <PW>
510 DATA▲PHAIP,EORIM,LSRIP,,JMPAB,EORAB,
LSRAB,,BVC▲R,EORIY,,,,EORZX,LSRZX,          <BU>
515 DATA▲CLIIP,EORAY,,,,EORAX,LSRAX,,RTS
IP,ADCIX,,,,ADCZP,RORZP,,PLAIP,ADCIM        <IQ>
520 DATA▲RORIP,,JMPIN,ADCAB,RORAB,,BVS▲R
,ADCIY,,,,ADCZX,RORZX,,SEIIP,ADCAY          <PP>
525 DATA▲,,,ADCAX,RORAX,,,STAIX,,,STYZP,
STAZP,STXZP,,DEYIP,,TXAIP,,STYAB,STAAB      <GO>
530 DATA▲STXAB,,BCC▲R,STAIY,,,STYZX,STAZ
X,STXZY,,TAYIP,STAAY,TXSIP,,,STAAX,,        <RR>
535 DATA▲LDYIM,LDAIX,LDXIM,,LDYZP,LDAZP,
LDXZP,,TAYIP,LDAIM,TAXIP,,LDYAB,LDAAB       <OG>
540 DATA▲LDXAB,,BCS▲R,LDAIY,,,LDYZX,LDAZ
X,LDXZY,,CLVIP,LDAAY,TSXIP,,LDYAX           <UR>
545 DATA▲LDAAX,LDXAY,,CPYIM,CMPIX,,,CPYZ
P,CMPZP,DECZP,,INYIP,CMPIM,DEXIP,,CPYAB     <AV>
550 DATA▲CMPAB,DECAB,,BNE▲R,CMPIY,,,,CMP
ZX,DECZX,,CLDIP,CMPAY,,,,CMPAX,DECAX,       <JL>
555 DATA▲CPXIM,SBCIX,,,CPXZP,SBCZP,INCZP
,,INXIP,SBCIM,NOPIP,,CPXAB,SBCAB,INCAB      <GX>
560 DATA▲,BEQ▲R,SBCIY,,,,SBCZX,INCZX,,SE
DIP,SBCAY,,,,SBCAX,INCAX,,,                 <VP>
565 --                                      <KH>
570 POSITION 7,10:? "Syntax▲-▲Error":GET
 X:GO# MAIN                                 <TI>
575 POSITION 7,10:? "Kein▲Drucker▲!!!":P
R=%0:GET X:GO# MAIN                         <TA>
580 --                                      <JW>
585 # MON:C=Z:POSITION %0,13:? "{ESC SHI
FT DEL}{ESC SHIFT DEL}"                     <RC>
590 FOR X=%0 TO 7:POSITION %0,22:? "$";H
EX$(C+X*8);"▲▲";:FOR I=%0 TO 7              <OG>
595 ? HEX$(PEEK(C+X*8+I));"▲";:NEXT I:?
:POSITION %0,13:? "{ESC SHIFT DEL}":NEXT
 X                                          <AR>
```

Listing. »ASS.PRG« ist der eigentliche Assembler (wird auf der nächsten Seite fortgesetzt)

```
600 POSITION %0,13:? "{ESC SHIFT DEL}":P
OSITION %0,22:? "Weiter▲mit▲SPACE▲,▲zuru
eck▲mit▲RETURN"                          <KQ>
605 DO :GET X:IF X<>32 AND X<>155 THEN L
OOP                                      <MK>
610 POP :POSITION %0,13:? "{ESC SHIFT DE
L}{ESC SHIFT DEL}":IF X=32 THEN C=C+64:G
OTO 590                                  <PQ>
615 GO# MAIN                             <EP>
620 --                                   <JL>
625 # DE:DPOKE Z+%1,VAL(G$(5)):B$=HEX$(P
EEK(Z+%1)):C$=HEX$(PEEK(Z+%2)):X=Z:Z=Z+%
3:GO# PRINT                              <JS>
630 --                                   <JN>
635 # ZE:POKE Z+%1,VAL(G$(6)):B$=HEX$(PE
EK(Z+%1)):X=Z:Z=Z+%2:GO# PRINT           <MD>
640 --                                   <JP>
645 # CHA:IF G$(5,5)<>"$" THEN G$(%2)=HE
X$(VAL(G$(5))):G$(%1,%1)="$":GOTO 655    <AD>
650 G$=STR$(DEC(G$(6)))                  <JM>
655 POSITION 7,10:? "entspricht▲▲";G$:GO
# MAIN                                   <UL>
660 --                                   <JT>
665 PROC INPUT:POSITION 7,7:? "▲▲▲▲▲▲▲▲▲▲
▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":POKE 752,%0:POSITION 6,7:
? "▲";:I=%0:G$=""                        <YV>
670 TRAP 670:GET X:IF (X>26 AND X<32) OR
 X=127 THEN 670                          <XY>
675 IF X=126 AND I=%0 THEN 670           <WS>
680 IF X=126 AND I=4 THEN IF G$(%1,%1)<>
";" THEN G$(4,5)="▲▲":I=I-%2:? "{ESC DEL
}{ESC DEL}";:GOTO 670                    <IW>
685 IF X=126 THEN G$(I,I)="▲":I=I-%1:? "
{ESC DEL}";:GOTO 670                     <KR>
690 IF X=155 THEN 715                    <OT>
695 I=I+%1:G$(I,I)=CHR$(X):? CHR$(X);    <VK>
700 IF I=%3 AND G$(%1,%1)<>";" THEN ? "▲
";:I=4:G$(I,I)="▲"                       <ND>
705 IF I=22 THEN 715                     <QO>
710 GOTO 670                             <QL>
715 IF LEN(G$)<>22 THEN G$(LEN(G$)+%1,22
)="▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲"                <NL>
720 POKE 752,%1:ENDPROC                  <FI>
725 --                                   <KB>
730 # FIL:MOVE Z,Z+%1,VAL(G$(5)):X=Z:Z=Z
+VAL(G$(5))+%1:GO# PRINT                 <MX>
735 --                                   <KD>
740 # GET:TRAP 745:OPEN #%1,4,%0,G$(5):%
GET #%1,X,I:BGET #%1,X,I:%GET #%1,Z:CLOS
E #%1:GO# MAIN                           <FY>
745 CLOSE #%1:POSITION 7,10:? "I/O▲-▲Err
or":GET X:GO# MAIN                       <BJ>
750 --                                   <JS>
755 # PUT:X=INSTR(G$,","):IF X=%0 THEN 5
70                                       <TB>
760 TRAP 745:I=VAL(G$(X+%1)):OPEN #%1,8,
%0,G$(5,X-%1):%PUT #%1,Z,I:BPUT #%1,Z,I:
%PUT #%1,Z:CLOSE #%1:GO# MAIN            <BP>
765 --                                   <KJ>
770 # DIR:POSITION %0,13:? "{ESC SHIFT D
EL}":TRAP 780:OPEN #%1,6,%0,"D:*.*"      <GV>
775 POSITION %0,13:? "{ESC SHIFT DEL}":I
NPUT #%1,G$:POSITION %0,22:? G$:PAUSE 10
:GOTO 775                                <BY>
780 CLOSE #%1:POSITION %0,13:? "{ESC SHI
FT DEL}":POKE 764,255:GO# MAIN           <FD>
785 --                                   <KN>
790 # ORG:IF G$(10,10)<>"▲" THEN 570     <UC>
795 Z=DEC(G$(6,9)):GO# MAIN              <MT>
```

Listing. »ASS.PRG« (Schluß)

Disketten-Quartett

Vier nützliche kleine Programme stellen wir Ihnen hier vor, mit denen Sie Files auf Diskette untersuchen, Disketten schnell formatieren und Ihr DOS an mehrere Laufwerke anpassen. Das vierte ist als Zusatz für unseren »CHAR-MAKER« gedacht, mit dem Sie dort erzeugte Zeichensätze laden.

Ein Diskettenlaufwerk ist eine feine Sache. Gerade, wenn man vorher eine ganze Weile mit dem Kassettenlaufwerk der Atari XL/XE-Computer gearbeitet hat, wird man von der Geschwindigkeit, mit der Daten geladen und gespeichert werden, berauscht sein. Nebenbei bietet die Diskettenstation noch einige andere Vorzüge. So braucht man sich überhaupt nicht um die Plazierung der Daten auf der Diskette zu kümmern. Zuverlässig sucht die Diskettenstation freien Speicherplatz auf der Diskette und findet genauso zuverlässig gespeicherte Daten oder Programme wieder.

Bald hat man sich an das Laufwerk so gewöhnt, daß man sich gar nicht vorstellen kann, je ohne dieses ausgekommen zu sein. So passen sich bald auch die geschriebenen Programme dem Betrieb mit der Diskettenstation an. Vier solcher Programme, die im Laufe der Zeit einfach dadurch entstanden sind, daß eine Diskettenstation im Computersystem zur Verfügung stand, wollen wir hier vorstellen. Alle Programme sind als Hexadezimal-Listings im AMPEL-Format abgedruckt und sollten mit diesem Eingabeprogramm abgetippt und unter dem im Listingkopf angegebenen Namen auf Diskette gespeichert werden.

ADDR zeigt den Speicherplatz

Das erste Programm ist dazu ausgelegt, die Speicherplatzbelegung eines Programms anzuzeigen. Im Gegensatz zu anderen Computern wird beim Atari ein Maschinensprache-Programm in mehrere kleine Programme aufgespaltet. Teile, die bei verschiedenen Programmen häufiger gebraucht werden, lassen sich so auf einfache Weise zu einem Programm dazukopieren.

Speichert man unter DOS 2.0 oder DOS 2.5 ein File mit der Funktion <K>, »Save File«, beziehungsweise unter OSS CP/A 1.2, 2.1, DOS XL oder Happy-DOS, mit »SAVE Filename Startadresse Endadresse« so wird am Anfang des Files die Start- und Endadresse des Files gespeichert. Das File sieht dann folgendermaßen aus: Die ersten beiden Byte haben den Wert 255 ($FF). Dadurch erkennt das DOS, daß es sich hier um ein File handelt, das mit dem »SAVE«-Kommando vom DOS aus gespeichert wurde und das sich mit dem »LOAD«-Befehl wieder laden läßt. Versucht man, ein File vom DOS aus zu laden, das diese Kennung nicht besitzt, beispielsweise ein Basic-Programm, so meldet das DOS nur »Bad Load File«. Ist die Kennung aber vorhanden, so weiß das DOS, daß die nächsten beiden Byte die Startadresse und die darauffolgenden Byte die Endadresse des Files darstellen. Die Bytes, die dann folgen, bestehen aus

den Daten des Files. Entweder ist das File dann zu Ende, oder es folgen wieder Start- und Endadresse, gefolgt von Datenbytes.

Diese Folge von Ladeadressen läßt sich mit dem ersten Programm »ADDR« auflisten. Sehr nützlich ist diese Funktion bei der Fehlersuche in eigenen Programmen, oder einfach nur, um in fremden Programmen herumzuschnüffeln. Wenden Sie das Programm doch einmal spaßeshalber auf den Turbo-Basic-Interpreter an. Sie werden staunen, aus wie vielen Teilen der Interpreter besteht.

Dabei werden Sie aber gleich noch etwas Bemerkenswertes feststellen. Beim Untersuchen von Turbo-Basic werden Sie zweimal auf die Zeile: »INIT AT ...« stoßen. Was hat das zu bedeuten?

Nun, in dem File lassen sich nicht nur die Start- und Endadresse eines Files festlegen. Gleichzeitig ist darin auch die Startadresse des Files gespeichert, wobei es sich dann um ein vom Mikroprozessor des Computers direkt ausführbares Maschinenprogramm handeln muß. Bei allen DOS-2-kompatiblen Operationssystemen, wie DOS 2.5, OSS CP/A 1.2, 2.1, DOS XL und Happy-DOS, gibt es zwei Arten von Startadressen. Die eine Adresse wird in den Bytes 738 und 739 gespeichert ($2E2 und $2E3) und bildet die Adresse, an der das File nach dem kompletten Laden gestartet wird. Die Startadresse des Programms legt man fest, indem man im File einfach eine Ladeadresse in diesen Speicherbereich angibt, gefolgt von zwei Byte, die die Startadresse des Programms enthalten.

Ein Programm soll zum Beispiel an der Adresse 8192 ($2000) gestartet werden. Also folgen im File vier Byte mit der Ladeadresse in die Speicherstellen 738 und 739: 226, 2, 227, 2 ($E2, $2, $E3, $2), in der Reihenfolge Low-Byte, High-Byte. Anschließend folgen zwei Byte, die in diesen Speicherbereich geschrieben werden und gleichzeitig die Startadresse des File bilden: 0, 32 ($0, $20). Wenn das File fertig geladen ist, springt das DOS automatisch zu dieser Adresse, und das Programm wird gestartet. Fehlt diese Ladeanweisung im File, werden die Daten des Files nur geladen.

DOS 2.0 und DOS 2.5 unterscheiden sich hier von den anderen DOS-Versionen. Unter CP/A 1.2, 2.1, DOS XL und Happy-DOS wird die Startadresse eines Files gleichgesetzt mit der ersten Ladeadresse des Files, solange keine andere Ladeadresse wie oben geschildert angegeben wird. Alle Files, bei denen unter diesen DOS-Versionen einfach nur der Name eingegeben wird und deren Extension »COM« lautet, werden automatisch gestartet (CP/M und MS-DOS lassen grüßen). Um den automatischen Start zu verhindern, gibt es bei diesen DOS-Versionen den »LOAD«-Befehl, über den einfach nur ein File in den Computerspeicher geladen, aber nicht gestartet wird.

Neben dieser normalen Startadresse gibt es noch eine weitere, die die normale Ladeprozedur des DOS beim »Binary Load«-Befehl unterbricht und kurzzeitig ein Programm ausführt, bevor das File weiter geladen wird. Diese Adresse wird in den Speicherstellen 736 und 737 zwischengespeichert ($2E0 und $2E2). Wird eine Adresse wie oben angegeben vom File in diese Speicherstellen geladen, so springt das DOS anschließend zu dieser Adresse und unterbricht damit kurzzeitig den Ladeprozeß. Erst wenn der Mikroprozessor in dem Programm auf den Befehl »RTS«, »Return from Subroutine«, stößt, wird der Ladeprozeß des Files vom DOS fortgeführt.

Bestes Beispiel für die Anwendung von Init-Adressen ist das Turbo-Basic. Dieses lädt in den Speicherbereich ab $5E00 die Grafik für das Happy-Computer-Zeichen und springt dann über eine Init-Adresse kurz zu einem Programm an Speicherstelle $2100, das das Ladebild erzeugt. Erst dann wird der Turbo-Basic-Interpreter fertig geladen.

Mit dem Programm ADDR lassen sich nun die Lade-, Start- und Init-Adressen von Files auf einfache Weise anzeigen. Nebenbei bemerkt, war das Programm eine große Hilfe, um das Turbo-Basic an den Kassettenrecorder-Betrieb anzupassen.

Falls Sie mehr als ein Laufwerk besitzen

Das zweite Programm ist dazu geschrieben worden, Ihr DOS mit mehreren Laufwerken zusammen arbeiten zu lassen. Um Daten vom und zum Laufwerk zu schicken, muß das DOS die Daten erst in einem Speicherbereich zwischenlagern. Immer wenn genug Daten zusammengekommen sind, werden diese auf die Diskette geschrieben oder von dieser heruntergeholt. Damit die Daten bei verschiedenen Laufwerken nicht durcheinanderkommen, hat jedes Laufwerk seinen eigenen vom DOS verwalteten Zwischenspeicher. Das gleiche gilt auch für die geöffneten Files. Jedem File wird Speicherplatz zugewiesen. Da an den Atari-Computer bis zu acht Laufwerke angeschlossen werden können, und da maximal acht Files gleichzeitig geöffnet werden können, würde immer ein riesengroßer Speicherplatz für die Laufwerke reserviert und vom übrigen Speicher getrennt werden. Da so mancher Atari-Besitzer froh ist, überhaupt ein Laufwerk zu besitzen, wäre dieser Speicherplatz in fast allen Fällen verschwendet. Aus diesem Grund läßt sich dem DOS die Anzahl der Laufwerke sowie die der maximal gleichzeitig geöffneten Files mitteilen.

Normalerweise geschieht dies vom Basic aus, mit umständlichen »POKE«-Befehlen. Viel zu schnell geht da mal ein POKE daneben und verdirbt dadurch das DOS. Wenn dies nicht bemerkt wird, ist beim nächsten Schreib- oder Lesebefehl eine Diskette hinüber.

Diese ganzen Nachteile umgeht der »DOS-Customizer«. Über eine Menüsteuerung lassen sich bequem die zur Verfügung stehenden Laufwerke sowie die maximal geöffneten Files einstellen. Hat man die Einstellungen vorgenommen, sollte man als nächstes diese angepaßte DOS-Version auf die Diskette schreiben. Dadurch übernimmt das DOS die Einstellungen beim nächsten Laden. Falls dann einmal ein zweites Laufwerk dazugekauft werden soll, läßt sich das DOS mit dem Customizer ohne Probleme wieder anpassen.

Der DOS-Customizer funktioniert nur mit dem DOS 2.0, DOS 2.5 und CP/A 1.2, 2.1 sowie DOS XL. Er sollte auf keinen Fall mit DOS 3 oder Happy-DOS verwendet werden.

Ein Dutzend Disketten formatiert

Haben Sie schon einmal mehrere Disketten hintereinander formatieren müssen, zum Beispiel nach einer schon lange fälligen Aufräumaktion in Ihrer Software-Sammlung? Womöglich hatten Sie sich ein neues Paket mit Disketten gekauft und wollten diese gleich mit formatieren. Wenn Sie dann vom DOS zum x-ten Mal nach dem Laufwerk, auf dem formatiert werden soll, gefragt wurden, und ob Sie sich auch ganz sicher sind, die Diskette ohne jeden Vorbehalt und ohne Rücksicht auf Verluste formatieren zu wollen, und ob Sie nicht vielleicht doch eine falsche Diskette eingelegt haben, und ..., und ..., und ..., und ..., wenn Ihnen dann schon einmal der Kragen geplatzt ist und Sie die DOS-Diskette wutschnaubend in die Ecke gefeuert haben, ja dann kommt unser Programm »Formater« für Sie genau richtig.

Mit diesem Programm läßt sich mit nur einem Knopfdruck eine Diskette formatieren und auch gleich ein DOS schreiben. Falls sich auf der Diskette kein DOS befindet, meldet diese anstelle eines nervös piepsenden »BOOT ERROR« einfach »Kein Dos auf dieser Diskette«. Wenn das nicht komfortabel ist?

Directory in Kyrillisch

Das letzte Programm kann man mehr als eine nette Spielerei betrachten. Das Programm kopiert bei den XL/XE-Computern einen beliebigen von der Diskette ladbaren Zeichensatz, zum Beispiel einen des in diesem Heft vorgestellten »CHAR-MAKER«, und verwendet diesen fortan zur Darstellung auf dem Bildschirm. Bis zum nächsten Druck auf die <RESET>-Taste haben Sie zum Beispiel Ihr Directory in kyrillischer Schrift oder Ihr Basic-Programm in Gotisch. So lassen sich auch ganz einfach die deutschen Umlaute sowie das »ß«, in eigene Programme einbinden, das im zweiten internationalen Zeichensatz des Ataris XL/XE nicht enthalten ist.

An dieser Stelle wollen wir Sie auffordern, uns ähnlich kurze, sinnvolle oder auch weniger sinnvolle, dafür aber nette Programme zu schicken. Diese vier Programme sollen als Anregung dienen. Gleichzeitig wünschen wir Ihnen mit unserem Programmquartett viel Spaß.

(Andreas Sons/hf)

```
Programmname :ADDR.COM
Länge        :888 Bytes
---------------------------------

0000:FF FF 00 20 FB 20 BA 86 <05>
0008:84 A9 9E 8D 44 03 A9 22 <D2>
0010:8D 45 03 A9 55 8D 48 03 <85>
0018:A9 00 8D 49 03 A9 0B 8D <80>
0020:42 03 A2 00 20 56 E4 A9 <1D>
0028:F3 8D 44 03 A9 22 8D 45 <A9>
0030:03 A9 0A 8D 48 03 A9 00 <7F>
0038:8D 49 03 A9 0B 8D 42 03 <22>
0040:A2 00 20 56 E4 A2 00 A9 <C9>
0048:05 8D 42 03 A9 4F 8D 44 <82>
0050:03 A9 23 8D 45 03 A9 0F <EF>
0058:8D 48 03 A9 00 8D 49 03 <71>
0060:20 56 E4 10 03 4C 4E 22 <47>
0068:AD 4F 23 C9 9B D0 04 A6 <09>
0070:84 9A 60 A2 10 A9 03 9D <74>
0078:42 03 A9 4C 9D 44 03 A9 <48>
0080:23 9D 45 03 A9 04 9D 4A <12>
0088:03 20 5F 22 20 2E 22 C9 <D5>
0090:00 F0 2C C9 84 F0 53 C9 <FA>
0098:FF F0 7A 20 24 22 A9 FD <3D>
00A0:8D 44 03 A9 22 8D 45 03 <38>
00A8:A9 0D 8D 48 03 A9 00 8D <FE>
00B0:49 03 A9 0B 8D 42 03 A2 <F1>
00B8:00 20 56 E4 4C 21 20 20 <CC>
00C0:2E 22 C9 00 D0 D5 20 24 <D3>
00C8:22 A9 0A 8D 44 03 A9 23 <BA>
00D0:8D 45 03 A9 0F 8D 48 03 <8F>
00D8:A9 00 8D 49 03 A9 0B 8D <C1>
00E0:42 03 A2 00 20 56 E4 4C <00>
00E8:21 20 20 2E 22 C9 09 D0 <3E>
00F0:AA A9 19 8D 44 03 A9 23 <D0>
00F8:8D 45 03 A9 0F 8D 48 03 <87>
0100:A9 00 FC 20 F7 21 8D 49 <F8>
0108:03 A9 0B 8D 42 03 A2 00 <83>
0110:20 56 E4 20 24 22 4C 21 <97>
0118:20 20 2E 22 C9 FF F0 03 <3F>
0120:4C 95 20 A9 28 8D 44 03 <4D>
0128:A9 23 8D 45 03 A9 0F 8D <F3>
0130:48 03 A9 00 8D 49 03 A9 <64>
0138:0B 8D 42 03 A2 00 20 56 <FE>
0140:E4 20 65 22 E0 E0 D0 39 <42>
0148:C0 02 D0 35 A9 37 8D 44 <79>
0150:03 A9 23 8D 45 03 A9 09 <EB>
0158:8D 48 03 A9 00 8D 49 03 <77>
0160:A9 0B 8D 42 03 A2 00 20 <ED>
0168:56 E4 20 65 22 20 65 22 <E4>
0170:86 80 98 20 00 22 A5 80 <5D>
0178:20 00 22 20 78 22 4C 37 <5C>
0180:21 E0 E2 D0 25 C0 02 D0 <40>
0188:21 A9 40 8D 44 03 A9 23 <B2>
0190:8D 45 03 A9 0A 8D 48 03 <AB>
0198:A9 00 8D 49 03 A9 0B 8D <F3>
01A0:42 03 A2 00 20 56 E4 4C <46>
01A8:60 21 86 80 84 81 98 20 <6F>
01B0:00 22 A5 80 20 00 22 A9 <AE>
01B8:4A 8D 44 03 A9 23 8D 45 <EB>
01C0:03 A9 01 8D 48 03 A9 00 <2F>
01C8:8D 49 03 A9 0B 8D 42 03 <70>
01D0:A2 00 20 56 E4 20 65 22 <99>
01D8:86 82 84 83 98 20 00 22 <32>
01E0:A5 82 20 00 22 20 2E 22 <A9>
01E8:A5 80 C5 82 D0 0C A5 81 <58>
01F0:C5 83 D0 06 20 78 22 4C <03>
01F8:37 21 18 A5 80 69 01 85 <C8>
0200:80 A5 F8 21 F3 22 81 69 <4A>
0208:00 85 81 4C DB 21 85 89 <3F>
0210:29 F0 4A 4A 4A 4A 20 0F <4B>
0218:22 A5 89 29 0F C9 0A 90 <39>
0220:03 18 69 07 18 69 30 AA <B8>
0228:AD 07 E4 48 AD 06 E4 48 <D2>
0230:8A 60 A2 10 A9 0C 9D 42 <77>
0238:03 4C 56 E4 A2 10 A9 07 <47>
0240:9D 42 03 A9 00 9D 44 03 <1C>
0248:9D 45 03 A9 01 9D 48 03 <E4>
0250:A9 00 9D 49 03 20 56 E4 <08>
0258:08 A5 00 28 30 01 60 20 <70>
0260:24 22 68 68 20 78 22 20 <A5>
0268:78 22 4C 21 20 20 56 E4 <5C>
0270:4C 4E 22 20 2E 22 48 20 <9C>
0278:2E 22 A8 68 AA E0 FF D0 <B5>
0280:04 C0 FF F0 EE 60 A9 4B <24>
0288:8D 44 03 A9 23 8D 45 03 <34>
0290:A9 01 8D 48 03 A9 00 8D <27>
0298:49 03 A9 0B 8D 42 03 A2 <FD>
02A0:00 20 56 E4 60 A5 88 48 <9E>
02A8:A5 87 48 60 9B 9B 9B 41 <E1>
02B0:44 44 52 20 20 20 20 56 <4B>
02B8:20 32 2E 30 20 2F 20 32 <40>
02C0:35 2D 41 55 47 2D 31 39 <B2>
02C8:38 35 9B 20 20 20 20 20 <6D>
02D0:20 20 20 56 20 32 2E 31 <A9>
02D8:20 2F 20 32 34 2D 4A 55 <17>
02E0:4E 2D 31 39 38 37 9B 9B <EF>
02E8:57 52 49 54 54 45 4E 20 <8F>
02F0:42 59 20 41 4E 44 52 45 <5F>
02F8:41 53 20 53 4F 4E 53 9B <20>
0300:9B 46 F4 22 5E 23 49 4C <5A>
0308:45 4E 41 4D 45 20 3E 46 <5C>
0310:4F 52 4D 41 54 20 3A 20 <7F>
0318:3F 3F 3F 9B 46 4F 52 4D <47>
0320:41 54 20 3A 20 42 41 53 <52>
0328:49 43 9B 46 4F 52 4D 41 <1A>
0330:54 20 3A 20 44 4F 53 20 <C3>
0338:31 9B 46 4F 52 4D 41 54 <F9>
0340:20 3A 20 44 4F 53 20 32 <63>
0348:9B 52 55 4E 20 41 54 20 <84>
0350:3A 20 49 4E 49 54 20 41 <82>
0358:54 20 3A 20 2D 9B 44 31 <E6>
0360:3A 20 20 20 20 20 20 20 <E2>
0368:20 20 20 20 20 20 20 20 <2C>
0370:20 00 00 00 00 00 00 00 <F5>
```

Listing 1. »ADDR« zeigt die Speicherbelegung

```
Programmname :DOSCUST.COM
Länge        :512 Bytes
---------------------------------

0000:FF FF 00 20 FB 20 BA 8E <0D>
0008:F5 21 A2 3C A0 21 A9 43 <EE>
0010:20 CC 20 A2 7F A0 21 A9 <9A>
0018:1E 20 CC 20 20 E6 20 AD <30>
0020:F6 21 C9 31 F0 23 C9 32 <7F>
0028:F0 1F C9 33 F0 1B C9 34 <E3>
0030:F0 17 C9 35 F0 13 C9 36 <EF>
0038:F0 0F C9 37 F0 0B C9 38 <EB>
0040:F0 07 A2 9D A0 21 4C 17 <27>
0048:21 8D F4 21 A2 B2 A0 21 <02>
0050:A9 17 20 CC 20 20 E6 20 <98>
0058:AD F6 21 C9 31 F0 13 C9 <25>
0060:32 F0 0F C9 33 F0 0B C9 <1C>
0068:34 F0 07 A2 C9 A0 21 4C <94>
0070:17 21 38 E9 31 AA BD 28 <6B>
0078:21 8D F3 21 A2 DE A0 21 <41>
0080:A9 15 20 CC 20 20 E6 20 <C8>
0088:AD F6 21 C9 59 F0 10 C9 <E9>
0090:4A F0 0C C9 4E D0 E5 A9 <EC>
0098:50 8D 79 07 4C 9E 20 A9 <85>
00A0:57 8D 79 07 AD F4 21 8D <53>
00A8:09 07 AD F3 21 8D 0A 07 <0E>
00B0:A9 FF 8D FC 02 4C 74 E4 <42>
00B8:8E 44 03 8C 45 03 A9 09 <8C>
00C0:8D 42 03 A9 FF 8D 48 03 <40>
00C8:8D 49 03 20 12 21 20 06 <1F>
00D0:21 60 8E 44 03 8C 45 03 <41>
00D8:8D 48 03 A9 00 8D 49 03 <F7>
00E0:A9 0B 8D 42 03 20 12 21 <8B>
00E8:20 06 21 60 A9 F6 8D 44 <0E>
00F0:03 A9 21 8D 45 03 A9 03 <04>
00F8:8D 48 03 A9 00 8D 49 03 <D7>
0100:A9 05 FC 20 F2 21 8D 42 <08>
0108:03 20 12 21 20 06 21 60 <9B>
0110:AD 43 03 C9 01 F0 04 AE <24>
0118:F5 21 9A 60 A2 00 4C 56 <11>
0120:E4 20 B2 20 A2 2C A0 21 <74>
0128:A9 10 20 CC 20 AE F5 21 <4A>
0130:9A 60 01 03 07 0F 50 52 <0A>
0138:4F 47 52 41 4D 20 41 42 <B1>
0140:4F 52 54 45 44 9B 44 4F <9F>
0148:53 20 43 55 53 54 4F 4D <6F>
0150:49 5A 45 52 9B 56 45 52 <29>
0158:53 49 4F 4E 20 31 2E 30 <E6>
0160:20 2F 20 32 31 2D 41 55 <59>
0168:47 2D 31 39 38 35 9B 57 <A5>
0170:52 49 54 54 45 4E 20 42 <1E>
0178:59 20 41 4E 44 52 45 41 <F2>
0180:53 20 53 4F 4E 53 9B 9B <13>
0188:9B 4D 41 58 49 4D 41 4C <A3>
0190:20 4E 55 4D 42 45 52 20 <7D>
0198:4F 46 20 4F 50 45 4E 20 <1C>
01A0:46 49 4C 45 53 20 3A 4D <AA>
01A8:55 53 54 20 42 45 54 57 <D3>
01B0:45 45 4E 20 31 20 41 4E <48>
01B8:44 20 38 9B 4E 55 4D 42 <31>
01C0:45 52 20 4F 46 20 44 49 <65>
01C8:53 4B 20 44 52 49 56 45 <F0>
01D0:53 20 3A 4D 55 53 54 20 <0C>
01D8:42 45 54 57 45 45 4E 20 <04>
01E0:31 20 41 4E 44 20 34 9B <E6>
01E8:56 45 52 49 46 59 20 57 <F0>
01F0:52 49 54 45 53 20 28 59 <8F>
01F8:2F 4E 29 20 3A 00 00 00 <C8>
```

Listing 2. »DOSCUST« paßt das DOS an

```
Programmname :FORMAT.COM
Länge        :944 Bytes
--------------------------------------

0000:FF FF 00 20 FB 20 A2 10 <5F>
0008:A9 0C 9D 42 03 20 56 E4 <BC>
0010:A2 20 A9 0C 9D 42 03 20 <73>
0018:56 E4 A2 10 A9 03 9D 42 <86>
0020:03 A9 8B 9D 44 03 A9 23 <A1>
0028:9D 45 03 A9 04 9D 4A 03 <1B>
0030:20 56 E4 A9 01 8D F0 02 <0D>
0038:A2 00 A9 0B 9D 42 03 A9 <87>
0040:36 9D 44 03 A9 22 9D 45 <87>
0048:03 A9 C0 9D 48 03 A9 00 <8F>
0050:9D 49 03 20 56 E4 BA 86 <8F>
0058:80 A2 00 A9 0B 9D 42 03 <6B>
0060:A9 89 9D 44 03 A9 23 9D <4C>
0068:45 03 A9 02 9D 48 03 A9 <3D>
0070:00 9D 49 03 20 56 E4 A2 <94>
0078:10 A9 07 9D 42 03 A9 00 <84>
0080:9D 44 03 9D 45 03 20 56 <67>
0088:E4 48 A2 00 A9 09 9D 42 <EF>
0090:03 A9 FF 9D 48 03 A9 00 <BF>
0098:9D 49 03 68 C9 1B D0 13 <FD>
00A0:A9 F6 9D 44 03 A9 22 9D <66>
00A8:45 03 20 56 E4 A9 00 8D <C8>
00B0:F0 02 60 C9 09 D0 16 A9 <1E>
00B8:08 9D 44 03 A9 23 9D 45 <EE>
00C0:03 20 56 E4 20 12 21 20 <FE>
00C8:4D 21 4C 53 20 C9 06 D0 <56>
00D0:16 A9 2A 9D 44 03 A9 23 <B7>
00D8:9D 45 03 20 56 E4 20 12 <6F>
00E0:21 20 BF 21 4C 53 20 C9 <1C>
00E8:17 D0 13 A9 3C 9D 44 03 <33>
00F0:A9 23 9D 45 03 20 56 E4 <1F>
00F8:20 4D 21 4C 53 20 C9 05 <66>
0100:F0 03 FC 20 F7 21 4C 71 <02>
0108:20 A9 4A 9D 44 03 A9 23 <F1>
0110:9D 45 03 20 56 E4 20 BF <D6>
0118:21 4C 53 20 A2 20 A9 FE <EE>
0120:9D 42 03 A9 8E 9D 44 03 <AA>
0128:A9 23 9D 45 03 20 56 E4 <B4>
0130:10 24 A2 00 A9 09 9D 42 <3D>
0138:03 A9 58 9D 44 03 A9 23 <6D>
0140:9D 45 03 A9 FF 9D 48 03 <DE>
0148:A9 00 9D 49 03 20 56 E4 <EE>
0150:A6 80 9A 4C 53 20 60 A2 <96>
0158:20 A9 24 9D 42 03 A9 8E <E0>
0160:9D 44 03 A9 23 9D 45 03 <95>
0168:20 56 E4 A2 20 A9 03 9D <AD>
0170:42 03 A9 8E 9D 44 03 A9 <6E>
0178:23 9D 45 03 A9 08 9D 4A <B4>
0180:03 20 56 E4 10 24 A2 00 <6B>
0188:A9 09 9D 42 03 A9 70 9D <B2>
0190:44 03 A9 23 9D 45 03 A9 <65>
0198:FF 9D 48 03 A9 00 9D 49 <4A>
01A0:03 20 56 E4 A6 80 9A 4C <E1>
01A8:53 20 A2 20 A9 0C 9D 42 <89>
01B0:03 20 56 E4 A2 20 A9 23 <27>
01B8:9D 42 03 A9 8E 9D 44 03 <F2>
01C0:A9 23 9D 45 03 20 56 E4 <ED>
01C8:60 A2 20 A9 24 9D 42 03 <0B>
01D0:A9 8E 9D 44 03 A9 23 9D <10>
01D8:45 03 20 56 E4 A2 20 A9 <16>
01E0:21 9D 42 03 A9 8E 9D 44 <F1>
01E8:03 A9 23 9D 45 03 20 56 <91>
01F0:E4 A9 57 8D 02 03 A9 04 <34>
01F8:8D 04 03 A9 22 8D 05 03 <8D>
0200:A9 01 F8 21 F3 22 8D 0A <6F>
0208:03 A9 00 8D 0B 03 20 53 <B8>
0210:E4 60 00 01 04 22 00 00 <D6>
0218:A2 00 A9 09 9D 42 03 A9 <B4>
0220:2B 9D 44 03 A9 22 9D 45 <ED>
0228:03 A9 FF 9D 48 03 A9 00 <21>
0230:9D 49 03 20 56 E4 4C 28 <30>
0238:22 4E 4F 20 44 4F 53 2E <FC>
0240:53 59 53 9B 9B 9B 9B 3E <D6>
0248:3E 4C 45 54 27 53 20 46 <59>
0250:4F 52 4D 41 54 3C 3C 9B <AB>
0258:9B 56 20 31 2E 30 20 2F <D3>
0260:20 32 33 2D 4A 55 4E 2D <69>
0268:31 39 38 37 20 62 79 20 <E8>
0270:41 6E 64 72 65 61 73 20 <87>
0278:53 6F 6E 73 9B 9B 50 72 <86>
0280:65 73 73 20 43 54 52 4C <7E>
0288:2D 46 20 74 6F 20 66 6F <2D>
0290:72 6D 61 74 9B 20 20 20 <50>
0298:20 20 20 43 54 52 4C 2D <9A>
02A0:57 20 74 6F 20 77 72 69 <E6>
02A8:74 65 20 44 4F 53 9B 20 <63>
02B0:20 20 20 20 20 43 54 52 <D7>
02B8:4C 2D 49 20 74 6F 20 66 <60>
02C0:6F 72 6D 61 74 20 61 6E <C9>
02C8:64 20 77 72 69 74 65 20 <DA>
02D0:44 4F 53 9B 20 20 20 20 <8C>
02D8:20 20 43 54 52 4C 2D 45 <F5>
02E0:20 74 6F 20 65 72 61 73 <DC>
02E8:65 20 44 4F 53 9B 20 20 <F9>
02F0:20 20 20 20 45 53 43 20 <B0>
02F8:20 20 20 74 6F 20 65 78 <14>
0300:69 74 F4 22 98 23 9B 9B <59>
0308:7E 7E 4C 65 61 76 69 6E <EC>
0310:67 20 70 72 6F 67 72 61 <49>
0318:6D 9B 7E 7E 46 6F 72 6D <83>
0320:61 74 74 69 6E 67 20 44 <8E>
0328:69 73 6B 20 61 6E 64 20 <B2>
0330:77 72 69 74 69 6E 67 20 <A4>
0338:44 4F 53 9B 7E 7E 46 6F <31>
0340:72 6D 61 74 74 69 6E 67 <71>
0348:20 44 69 73 6B 9B 7E 7E <02>
0350:57 72 69 74 69 6E 67 20 <54>
0358:44 4F 53 9B 7E 7E 45 72 <C4>
0360:61 73 69 6E 67 20 44 4F <F2>
0368:53 9B 45 72 72 6F 72 20 <7A>
0370:64 75 72 69 6E 67 20 66 <E1>
0378:6F 72 6D 61 74 74 69 6E <72>
0380:67 9B 45 72 72 6F 72 20 <F4>
0388:64 75 72 69 6E 67 20 77 <99>
0390:72 69 74 69 6E 67 20 44 <D8>
0398:4F 53 9B 3E 5F 4B 3A 9B <79>
03A0:44 31 3A 44 4F 53 2E 53 <03>
03A8:59 53 9B 00 00 00 00 00 <36>
```

Listing 3. Mit »FORMAT« sind Disketten ruck zuck formatiert

```
Programmname :XLCHAR.COM
Länge        :352 Bytes
--------------------------------------

0000:FF FF 00 20 FB 20 A2 00 <4F>
0008:20 2F 20 A2 00 8E 0E D2 <26>
0010:8E 0E D4 8E 00 D4 CA AD <CE>
0018:01 D3 29 FE 8D 01 D3 48 <B0>
0020:20 2F 20 A2 09 BD 6F 20 <52>
0028:9D 48 15 CA 10 F7 68 8D <C2>
0030:49 15 4C 79 20 86 C4 A2 <61>
0038:00 A9 00 85 C0 85 C2 A9 <FB>
0040:C0 85 C1 A9 50 85 C3 A5 <C3>
0048:C4 F0 07 A1 C2 81 C0 4C <F4>
0050:50 20 A1 C0 81 C2 E6 C0 <13>
0058:E6 C2 A5 C0 D0 E9 E6 C1 <4B>
0060:E6 C3 A5 C1 F0 0E C9 D2 <E0>
0068:F0 F4 C9 D3 F0 F0 C9 D4 <48>
0070:F0 EC D0 D3 60 A9 FC 8D <D5>
0078:01 D3 A9 00 8D C6 02 A9 <F3>
0080:84 85 0C A9 20 85 0D 4C <AD>
0088:74 E4 A9 40 85 0C A9 15 <CF>
0090:85 0D 20 E0 07 A2 00 A9 <04>
0098:0B 9D 42 03 A9 0B 9D 44 <AC>
00A0:03 A9 21 9D 45 03 A9 46 <19>
00A8:9D 48 03 A9 00 9D 49 03 <40>
00B0:20 56 E4 A9 05 9D 42 03 <8F>
00B8:A9 54 9D 44 03 A9 21 9D <B4>
00C0:45 03 A9 0F 9D 48 03 A9 <03>
00C8:00 9D 49 03 20 56 E4 AD <58>
00D0:49 15 8D 01 D3 A2 10 A9 <42>
00D8:03 9D 42 03 A9 51 9D 44 <07>
00E0:03 A9 21 9D 45 03 A9 04 <16>
00E8:9D 4A 03 20 56 E4 A9 07 <88>
00F0:9D 42 03 A9 00 9D 44 03 <78>
00F8:9D 48 03 A9 E0 9D 45 03 <41>
0100:A9 04 FC 20 53 21 9D 49 <F8>
0108:03 20 56 E4 A9 0C 9D 42 <8F>
0110:03 20 56 E4 60 9B 9B 58 <7F>
0118:4C 43 48 41 52 9B 56 45 <18>
0120:52 53 49 4F 4E 20 31 2E <B9>
0128:30 20 20 2F 20 20 32 38 <4F>
0130:2D 4F 43 54 2D 31 39 38 <FE>
0138:35 9B 57 52 49 54 54 45 <34>
0140:4E 20 42 59 20 41 4E 44 <2D>
0148:52 45 41 53 20 53 4F 4E <08>
0150:53 9B 9B 43 48 41 52 53 <87>
0158:45 54 3E 44 31 3A 00 00 <43>
```

Listing 4. Andere Zeichensätze mit »XCHAR«

DR. NIBBLE & crew



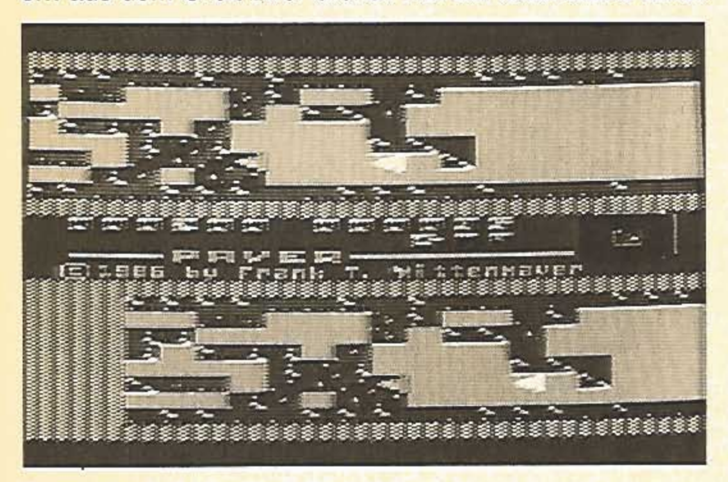




Asteroiden-Action

Was kann spannender sein als ein Spiel zu zweit? »Paver« verspricht nicht nur Spielspaß, sondern auch tolle Musik und butterweiches Scrolling auf einem geteilten Bildschirm.

Etwa im Jahr 2700 entdeckte die Menschheit, daß kleine, wendige Gleiter und die Schwerelosigkeit ganz neue Spiele erlauben. 2978 kam ein begeisterter Spieler auf eine völlig neue Idee: Zwei Spieler fernsteuern aus dem Orbit zwei Gleiter. Auf der Oberfläche eines



Spannung auf geteiltem Bildschirm

Asteroiden wird ein Spielfeld markiert, das aus lauter symmetrischen Teilen besteht. Dann wird ein Großteil des Feldes wieder mit Staub abgedeckt, nur am oberen und unteren Rand bleibt ein schmaler Streifen frei. Der Computer gibt im Laufe des Spiels einen Teil des Musters an, und die Spieler müssen zu der Stelle fliegen, an die dieses Muster gehört. Hat man sie erreicht, befreit der Computer sie automatisch vom Staub, der Spieler bekommt Punkte, abhängig von der Anzahl der Nachbarplatten, und die Jagd auf das nächste Teil beginnt.

Damit das Spiel nicht an Spannung verliert, kann man sich auch gegenseitig abschießen. Das bringt zwar keine Punkte, dafür dem Mitspieler einen Zeitnachteil. Er beginnt nämlich wieder am linken Ende des Spielfelds, was besonders am Ende des Spiels, wenn nur noch wenige Stellen übrig sind, sehr ärgerlich ist. Sie müssen sich also auf mehrere Dinge gleichzeitig konzentrieren.

Wenn Sie das Listing »Paver« mit AMPEL abgetippt haben, können Sie es über das DOS starten. Zum Spielen brauchen Sie unbedingt einen menschlichen Mitspieler und zwei Joysticks. Der Spieler mit dem Joystick in Port 2 steuert den rosa Gleiter auf dem unteren Bildschirmausschnitt. Bei Paver spielen Sie nämlich auf einem geteilten Bildschirm, wie man ihn zum Beispiel von »Ballblazer« her kennt. Die Schirme können völlig unterschiedliche Ausschnitte des großen Spielfelds zeigen. So sehen Sie nicht nur, was Ihr eigener Gleiter macht, sondern haben auch Ihren Gegner im Auge. Wenn Sie am hinteren Ende des Spielfelds angekommen sind, können Sie entweder umdrehen oder geradeaus weiterfliegen. Nach einer Weile kommen Sie dann wieder zum Anfang des Felds.

Solang noch die Titelmusik läuft, haben Sie die Wahl zwischen zwei Spielmodi: »GAP« und »NON-GAP«. Bei GAP ist das Spiel zu Ende, sobald einer der Spieler 1000 Punkte erreicht hat. Wenn Sie NON-GAP wählen, müssen Sie alle Felder wieder vom Staub befreien. Gewonnen hat dann der Spieler mit der höheren Punktzahl. Durch Drücken der <SELECT>-Taste können Sie vor jedem Spiel den Modus bestimmen. Um das Spiel zu beginnen, drücken Sie einfach die <START>-Taste.

(Frank Wittenmayer/gn/hf)

Steckbrief	
Programm:	Paver
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Kassette, Diskette

```
Programmname :PAVER.COM
Länge        :7568 Bytes
--------------------------------------

0000:FF FF D0 22 4F 40 00 3B <92>
0008:D0 22 DD 22 A9 3C 8D 02 <04>
0010:D3 18 60 A9 DD 85 02 85 <FC>
0018:0C 85 0A A9 22 85 03 85 <ED>
0020:0D 85 0B A9 00 8D 44 02 <6E>
0028:A9 02 85 09 4C 4A 37 01 <91>
0030:A5 C5 38 E9 01 85 95 A9 <79>
0038:00 85 94 A2 04 A0 FF 91 <13>
0040:94 88 D0 FB E6 95 CA 10 <FE>
0048:F6 AD 2F 02 09 1C 8D 2F <B5>
0050:02 A9 03 8D 1D D0 A9 01 <5F>
0058:8D 6F 02 A9 3A 8D 26 02 <97>
0060:A9 23 8D 27 02 A9 00 8D <B1>
0068:19 02 A9 01 8D 18 02 60 <CB>
0070:20 51 23 20 9B 23 20 0D <6D>
0078:24 20 98 2E 20 69 2D A9 <89>
0080:01 85 4D 8D 18 02 60 A2 <FE>
0088:03 B4 84 B9 15 31 48 B9 <E6>
0090:2A 31 9D C0 02 98 0A A8 <AE>
0098:B9 3B 31 85 80 B9 3C 31 <02>
00A0:85 81 B5 88 9D 00 D0 B5 <F4>
00A8:8C 85 82 B5 90 85 D2 8A <5D>
00B0:18 65 C5 85 83 85 D3 A0 <22>
00B8:14 A9 00 91 D2 88 10 FB <02>
00C0:68 A8 B1 80 91 82 88 10 <AA>
00C8:F9 B5 8C 95 90 CA 10 B9 <FA>
00D0:60 A2 03 A5 C5 38 E9 01 <AD>
00D8:85 83 85 D3 B4 A0 B9 15 <64>
00E0:31 48 98 0A A8 B9 3B 31 <CA>
00E8:85 80 B9 3C 31 85 81 B5 <37>
00F0:A4 9D 04 D0 B5 A8 85 82 <7A>
00F8:B5 AC 85 D2 A0 09 B1 D2 <80>
0100:3D E5 23 91 D2 88 10 F6 <0C>
0108:68 A8 B1 80 11 82 91 82 <88>
0110:88 10 F7 B5 A8 95 AC CA <37>
0118:10 C2 60 FC F3 CF 3F 48 <F7>
0120:8A 0A AA 68 95 CA 94 B0 <58>
0128:94 B1 A9 00 95 CB 60 A9 <0D>
0130:00 8D 08 D2 A9 03 8D 0F <EE>
0138:D2 A2 07 A9 00 95 CA CA <A6>
0140:10 FB 60 A5 BC F0 03 4C <5F>
0148:FB 25 A2 06 B5 CA F0 35 <87>
0150:0A A8 B9 65 24 85 C6 B9 <48>
0158:66 24 85 C7 B9 77 24 85 <83>
0160:C8 B9 78 24 85 C9 B4 CB <5F>
0168:B1 C8 F0 19 9D 01 D2 B1 <C2>
0170:C6 9D 00 D2 D6 B1 B5 B1 <69>
0178:10 06 B5 B0 95 B1 F6 CB <53>
0180:CA CA 10 C8 60 A9 00 95 <13>
0188:CA AD 0A D2 29 07 69 5A <0B>
0190:9D 00 D2 A9 81 9D 01 D2 <90>
0198:4C 4A 24 89 24 8C 24 B1 <AC>
01A0:24 C3 24 8A 24 AF 24 C1 <67>
01A8:24 E9 24 E7 24 89 24 9E <13>
01B0:24 9E 24 D5 24 9C 24 9C <C0>
01B8:24 D3 24 F7 24 F5 24 00 <B2>
01C0:00 00 E7 E9 EB ED EF F1 <28>
01C8:F3 F5 F7 F9 FA FB FC FD <B0>
01D0:FE FF A8 A8 A8 A6 A5 A5 <CB>
01D8:A4 A4 A4 A4 A4 A4 A3 A3 <DC>
01E0:A3 A3 A3 A2 00 00 00 83 <DE>
01E8:85 87 89 8B 8D 8F 91 93 <07>
01F0:95 96 97 98 99 9A 9B 00 <81>
01F8:00 03 05 0A 0A 04 04 04 <69>
0200:04 03 03 03 02 02 02 01 <74>
0208:01 0C 0C 0C 0F 0D 04 05 <8B>
0210:06 06 05 05 04 04 03 03 <C2>
0218:02 02 01 01 00 00 00 02 <CF>
0220:02 02 03 03 03 04 04 04 <6A>
0228:05 05 05 01 01 0F 0D 0B <09>
0230:09 07 05 04 03 02 02 01 <80>
0238:01 00 00 00 00 00 00 00 <BC>
0240:00 00 00 00 00 00 00 00 <44>
0248:00 00 00 00 00 00 00 00 <4C>
0250:00 00 00 00 00 00 00 00 <54>
0258:00 00 00 00 00 00 00 00 <5C>
0260:00 00 00 00 00 00 00 00 <64>
0268:00 00 00 BD 1C 25 D0 0A <57>
0270:9D 00 D2 9D 01 D2 FE 24 <56>
0278:25 60 9D 00 D2 BD 24 25 <4B>
0280:C9 03 90 17 BD 14 25 38 <64>
0288:FD 24 25 C9 04 90 0A BD <78>
0290:2C 25 9D 01 D2 FE 24 25 <8B>
0298:60 49 07 8D 34 25 FE 24 <1E>
02A0:25 BD 2D 25 0A 0A 0A 18 <02>
02A8:6D 34 25 A8 BD 2C 25 29 <A9>
```

```
02B0:F0 18 79 F9 26 9D 01 D2 <C8>
02B8:60 BD 04 25 18 69 02 9D <F2>
02C0:04 25 85 B8 BD 05 25 69 <D9>
02C8:00 9D 05 25 85 B9 A0 00 <6F>
02D0:B1 B8 C9 FF D0 09 FE 0C <CE>
02D8:25 20 B8 25 4C 83 25 9D <04>
02E0:1C 25 C8 B1 B8 9D 14 25 <C5>
02E8:A9 00 9D 24 25 60 BD F3 <46>
02F0:25 85 BA BD F4 25 85 BB <83>
02F8:BC 0C 25 B1 BA C9 FF D0 <3C>
0300:08 A9 00 9D 0C 25 4C C2 <9D>
0308:25 0A A8 B9 11 27 85 BA <AD>
0310:9D 04 25 B9 12 27 85 BB <07>
0318:9D 05 25 A0 00 B1 BA 9D <B4>
0320:2C 25 C8 B1 BA 9D 2D 25 <2F>
0328:60 62 26 88 26 AD 26 D2 <36>
0330:26 A2 06 BD 24 25 DD 14 <0F>
0338:25 D0 03 20 83 25 20 35 <43>
0340:25 CA CA 10 EE 60 A2 06 <C1>
0348:A9 00 9D 0C 25 A8 BD F3 <47>
0350:25 85 BA BD F4 25 85 BB <15>
0358:B1 BA 0A A8 B9 11 27 85 <C6>
0360:BA B9 12 27 85 BB A0 00 <B4>
0368:B1 BA 9D 2C 25 C8 B1 BA <31>
0370:9D 2D 25 C8 B1 BA 9D 1C <FA>
0378:25 C8 B1 BA 9D 14 25 A9 <7C>
0380:00 9D 24 25 A5 BA 18 69 <6A>
0388:02 9D 04 25 A5 BB 69 00 <B2>
0390:9D 05 25 CA CA 10 B1 60 <39>
0398:00 01 02 03 00 01 02 04 <5B>
03A0:05 06 07 08 09 06 07 08 <82>
03A8:09 06 07 08 0A 0B 0C 0D <B7>
03B0:0E 06 07 08 09 06 07 08 <F6>
03B8:0A 0B 0C 0D 0F FF 10 11 <7E>
03C0:12 13 10 11 12 13 10 11 <AD>
03C8:12 13 10 11 12 13 10 11 <B5>
03D0:12 14 15 16 10 17 10 11 <F6>
03D8:12 13 10 11 12 18 15 16 <E8>
03E0:10 17 FF 19 19 19 19 19 <9E>
03E8:19 19 19 19 19 19 19 19 <AE>
03F0:19 19 19 19 19 19 19 19 <76>
03F8:19 19 1A 19 19 19 19 19 <9E>
0400:19 19 19 19 19 19 1A FF <E7>
0408:1B 00 01 02 03 00 01 02 <F9>
0410:04 05 06 07 08 09 06 07 <03>
0418:08 09 06 07 08 0A 0B 0C <20>
0420:0D 0E 06 07 08 09 06 07 <DA>
0428:08 0A 0B 0C 0D 1C FF 00 <AF>
0430:01 02 04 04 02 01 00 00 <0E>
0438:00 01 02 02 01 00 00 03 <EB>
0440:02 01 00 00 00 00 00 4B <D4>
0448:27 64 27 7D 27 96 27 B1 <28>
0450:27 C6 27 CF 27 D4 27 E1 <D2>
0458:27 E6 27 F3 27 FA 27 13 <03>
0460:28 2A 28 43 28 50 28 57 <21>
0468:28 70 28 89 28 A2 28 BD <92>
0470:28 DA 28 F3 28 0A 29 11 <7F>
0478:29 2C 29 3D 29 44 29 49 <0E>
0480:29 C4 00 30 10 28 10 24 <66>
0488:10 37 08 30 08 30 08 28 <A3>
0490:10 24 08 24 10 37 08 33 <8D>
0498:08 FF C4 00 30 10 28 10 <1F>
04A0:24 10 37 08 30 08 30 08 <0E>
04A8:28 10 24 08 24 10 30 08 <97>
04B0:2D 08 FF C4 00 2A 10 24 <6A>
04B8:10 1F 10 30 08 2A 08 2A <B8>
04C0:08 24 10 1F 08 1F 10 30 <D6>
04C8:08 2D 08 FF C4 00 2A 10 <9B>
04D0:24 10 1F 10 30 08 2A 08 <D1>
04D8:2A 08 24 10 1F 08 1F 08 <DC>
04E0:2A 08 1F 08 2A 08 FF C4 <77>
04E8:00 2A 10 24 10 1F 10 30 <04>
04F0:08 2A 08 2A 08 24 10 1F <FA>
04F8:08 1F 08 FF A6 00 51 08 <92>
0500:3C 08 51 08 FF A6 00 4C <B3>
0508:80 FF A6 00 4C 60 4C 08 <43>
0510:4C 08 3C 08 55 08 FF A6 <B9>
0518:00 51 80 FF A6 00 51 60 <84>
0520:51 08 51 08 3C 08 51 08 <22>
0528:FF A6 00 51 60 00 20 FF <62>
0530:A6 00 79 10 66 10 5B 10 <6F>
0538:79 08 88 08 88 08 72 10 <AB>
0540:66 08 66 10 88 08 90 08 <B0>
0548:FF A6 00 99 10 80 10 72 <19>
0550:10 99 08 A2 08 A2 08 88 <02>
0558:10 79 08 AD 10 79 10 FF <D9>
0560:A6 00 B6 10 99 10 88 10 <19>
0568:CC 08 B6 08 B6 08 99 10 <A8>
0570:88 08 88 10 CC 08 C1 08 <1A>
0578:FF A6 00 B6 20 00 48 51 <E1>
0580:08 3C 08 51 08 FF A6 00 <2B>
0588:B6 20 00 60 FF C3 01 61 <62>
0590:10 52 10 48 10 70 08 61 <6F>
0598:08 61 08 52 10 48 08 48 <0E>
05A0:10 70 08 66 08 FF C3 01 <AA>
05A8:61 10 52 10 48 10 70 08 <1C>
05B0:61 08 61 08 52 10 48 08 <85>
05B8:48 10 61 08 5A 08 FF C3 <43>
05C0:01 55 10 48 10 3F 10 61 <21>
05C8:08 55 08 55 08 48 10 3F <32>
05D0:08 3F 10 61 08 5A 08 FF <7B>
05D8:C3 01 55 10 48 10 3F 10 <31>
05E0:61 08 55 08 55 08 48 10 <93>
05E8:3F 08 3F 08 55 08 3F 08 <CD>
05F0:55 08 FF C3 01 55 08 55 <F1>
05F8:08 48 10 3F 10 61 08 55 <75>
0600:08 55 08 48 10 3F 08 3F <B3>
0608:08 55 08 48 08 43 08 FF <3B>
0610:C3 01 3F 10 37 10 30 10 <69>
0618:3F 08 48 08 48 08 3C 10 <30>
0620:37 08 37 10 48 08 4B 08 <AE>
0628:FF C3 01 52 10 43 10 3C <93>
0630:10 52 08 55 08 55 08 48 <18>
0638:10 3F 08 55 10 3F 10 FF <FE>
0640:C3 01 61 20 00 60 FF C3 <5E>
0648:01 55 10 48 10 3F 10 61 <9F>
0650:08 55 08 55 08 48 10 3F <CC>
0658:08 3F 08 55 08 48 08 43 <42>
0660:08 FF 00 02 0A 10 0A 10 <04>
0668:0A 20 0A 08 0A 10 0A 08 <AF>
0670:0A 20 FF 00 02 0A 08 0A <3B>
0678:78 FF A6 00 00 04 FF A6 <09>
0680:00 B6 20 00 5C FF 20 5F <B4>
0688:2D 20 6E 29 20 AC 2C 20 <9F>
0690:B7 2C 20 60 29 60 A9 A0 <07>
0698:8D 07 D4 18 69 04 85 C5 <54>
06A0:20 FA 22 60 A9 00 8D 30 <05>
06A8:02 A9 2C 8D 31 02 A9 3C <3D>
06B0:8D F4 02 A9 36 8D C4 02 <4C>
06B8:A9 34 8D C5 02 A9 08 8D <77>
06C0:C6 02 A9 1F 8D C7 02 A9 <63>
06C8:48 8D 22 2C A9 2C 8D 23 <35>
06D0:2C 60 00 00 00 00 00 00 <CA>
06D8:00 00 00 00 00 00 00 00 <E4>
06E0:00 00 00 00 00 00 00 00 <EC>
06E8:00 00 00 00 00 00 00 00 <F4>
06F0:00 00 00 00 00 00 00 00 <FC>
06F8:00 00 00 00 00 00 00 00 <04>
0700:00 00 00 00 00 00 00 00 <0E>
0708:00 00 00 00 00 00 00 00 <16>
0710:00 00 00 00 00 00 00 00 <1E>
0718:00 00 00 00 00 00 00 00 <26>
0720:00 00 00 00 00 00 00 00 <2E>
0728:00 00 00 00 00 00 00 00 <36>
0730:00 00 00 00 00 00 00 00 <3E>
0738:00 00 00 00 00 00 00 00 <46>
0740:00 00 00 00 00 00 00 00 <4E>
0748:00 00 00 00 00 00 00 00 <56>
0750:00 00 00 00 00 00 00 00 <5E>
0758:00 00 00 00 00 00 00 00 <66>
0760:00 00 00 00 00 00 00 00 <6E>
0768:00 00 00 00 00 00 00 00 <76>
0770:00 00 00 00 00 00 00 00 <7E>
0778:00 00 00 00 00 00 00 00 <86>
0780:00 00 00 00 00 00 00 00 <8E>
0788:00 00 00 00 00 00 00 00 <96>
0790:00 00 00 00 00 00 00 00 <9E>
0798:00 00 00 00 00 00 00 00 <A6>
07A0:00 00 00 00 00 00 00 00 <AE>
07A8:00 00 00 00 00 00 00 00 <B6>
07B0:00 00 00 00 00 00 00 00 <BE>
07B8:00 00 00 00 00 00 00 00 <C6>
07C0:00 00 00 00 00 00 00 00 <CE>
07C8:00 00 00 00 00 00 00 00 <D6>
07D0:00 00 00 00 00 00 00 00 <DE>
07D8:00 00 00 00 00 00 00 00 <E6>
07E0:00 00 00 00 00 00 00 00 <EE>
07E8:00 00 00 00 00 00 00 00 <F6>
07F0:00 00 00 00 00 00 00 00 <FE>
07F8:00 00 00 00 00 00 00 00 <06>
0800:00 00 00 00 00 00 00 00 <10>
0808:00 00 00 00 00 00 00 00 <18>
0810:00 00 00 00 00 00 00 00 <20>
0818:00 00 00 00 00 00 00 00 <28>
0820:00 00 00 00 00 00 00 00 <30>
0828:00 00 00 00 00 00 00 00 <38>
0830:00 00 00 00 00 00 00 00 <40>
0838:00 00 00 00 00 00 00 00 <48>
0840:00 00 00 00 00 00 00 00 <50>
0848:00 00 00 00 00 00 00 00 <58>
0850:00 00 00 00 00 00 00 00 <60>
0858:00 00 00 00 00 00 00 00 <68>
0860:00 00 00 00 00 00 00 00 <70>
0868:00 00 00 00 00 00 00 00 <78>
0870:00 00 00 00 00 00 00 00 <80>
0878:00 00 00 00 00 00 00 00 <88>
0880:00 00 00 00 00 00 00 00 <90>
0888:00 00 00 00 00 00 00 00 <98>
0890:00 00 00 00 00 00 00 00 <A0>
0898:00 00 00 00 00 00 00 00 <A8>
08A0:00 00 00 00 00 00 00 00 <B0>
08A8:00 00 00 00 00 00 00 00 <B8>
08B0:00 00 00 00 00 00 00 00 <C0>
08B8:00 00 00 00 00 00 00 00 <C8>
08C0:00 00 00 00 00 00 00 00 <D0>
08C8:00 00 00 00 00 00 00 00 <D8>
08D0:00 00 00 00 00 00 00 00 <E0>
08D8:00 00 00 00 00 00 00 00 <E8>
08E0:00 00 00 00 00 00 00 00 <F0>
08E8:00 00 00 00 00 00 00 00 <F8>
08F0:00 00 00 00 00 00 00 00 <00>
08F8:00 00 00 00 00 00 00 00 <08>
0900:00 00 00 00 00 00 00 00 <12>
0908:00 00 00 00 00 00 00 00 <1A>
0910:00 00 00 00 00 00 00 00 <22>
0918:00 00 00 00 00 00 00 00 <2A>
0920:00 00 00 00 00 00 00 00 <32>
0928:00 00 00 00 00 00 00 00 <3A>
0930:00 00 00 00 00 00 70 70 <93>
0938:F0 56 00 80 56 00 81 56 <E3>
0940:00 82 56 00 83 56 00 84 <AF>
0948:56 00 85 56 00 86 56 00 <DD>
0950:87 56 00 88 56 00 89 46 <BF>
0958:48 2C 06 86 04 56 00 80 <2B>
0960:56 00 81 56 00 82 56 00 <35>
0968:83 56 00 84 56 00 85 56 <6D>
0970:00 86 56 00 87 56 00 88 <00>
0978:56 00 89 41 00 2C 00 10 <9B>
0980:10 10 10 10 10 00 10 10 <32>
0988:10 10 10 10 00 00 DD CF <32>
0990:DE 00 00 00 00 00 00 00 <91>
0998:00 00 00 00 00 00 00 00 <AA>
09A0:00 00 CF 00 CF 00 00 9B <9E>
09A8:9A 9A B0 A1 B6 A5 B2 9B <7E>
09B0:9A 9A 9A 9A 9A 9C FE CF <11>
09B8:DF 00 00 00 3C 3D 11 19 <4C>
09C0:18 16 00 62 79 00 26 72 <14>
09C8:61 6E 6B 00 34 3B 00 37 <D9>
09D0:69 74 74 65 6E 6D 61 79 <5D>
09D8:65 72 00 00 00 00 00 00 <B9>
09E0:00 00 A9 00 8D 08 D2 A9 <E1>
09E8:03 8D 0F D2 60 A0 00 A9 <B1>
09F0:00 85 DF A9 80 85 E0 4C <0D>
09F8:C6 2C C5 2C AD C5 2C 48 <EC>
0A00:AD C4 2C 48 A9 3F 91 DF <06>
0A08:A5 DF 18 69 01 85 DF A5 <B1>
0A10:E0 69 00 85 E0 A5 E0 C9 <D1>
0A18:8A D0 04 A5 DF C9 00 F0 <42>
0A20:01 60 68 68 A0 40 A9 BF <F8>
0A28:99 00 82 99 00 83 99 00 <9F>
0A30:84 99 00 85 99 00 86 99 <1D>
0A38:00 87 AD 0A D2 29 05 09 <AA>
0A40:4A 99 00 81 AD 0A D2 29 <01>
0A48:05 09 4A 99 00 88 C8 C0 <63>
0A50:7E 90 D3 A9 FF 85 C4 A2 <31>
0A58:04 A9 10 9D 49 2C 9D 50 <E8>
0A60:2C CA 10 F7 A9 2F 85 D5 <0C>
0A68:85 D7 A9 18 85 D8 A9 30 <19>
0A70:85 D9 A9 00 85 E3 85 E5 <7A>
0A78:85 EC 85 ED A9 03 85 84 <B0>
0A80:85 86 A9 0B 85 85 85 87 <86>
0A88:A9 0F 85 DA 85 DB A9 07 <53>
0A90:85 EA 85 EB 60 A2 7F A9 <76>
0A98:00 95 80 CA 10 FB 60 20 <BB>
0AA0:9B 2D 20 CE 2D 20 82 2D <56>
0AA8:20 8F 32 20 66 2F 20 AE <D7>
0AB0:2E 20 F0 2F 20 7E 35 60 <98>
0AB8:A5 D5 A0 1B 99 04 2C 88 <B7>
0AC0:88 88 10 F8 A5 D7 A0 1B <18>
0AC8:99 28 2C 88 88 88 10 F8 <AF>
0AD0:60 A5 EC D0 15 A5 D4 D0 <2F>
0AD8:11 A5 D8 29 07 D0 0B AD <6A>
0AE0:78 02 85 DA C9 0F F0 02 <D3>
0AE8:85 EA A5 ED D0 15 A5 D6 <0F>
0AF0:D0 11 A5 D9 29 07 D0 0B <E4>
0AF8:AD 79 02 85 DB C9 0F F0 <A5>
0B00:02 85 EB 60 A0 01 B6 DA <22>
0B08:98 48 0A A8 B9 D4 00 18 <6D>
0B10:7D 34 2E 99 D4 00 B9 D5 <2E>
0B18:00 69 00 C9 88 90 02 A9 <00>
0B20:21 99 D5 00 B9 D4 00 38 <14>
0B28:FD 24 2E 99 D4 00 B9 D5 <B9>
0B30:00 E9 00 C9 20 B0 02 A9 <89>
0B38:87 99 D5 00 68 A8 B9 D8 <BF>
0B40:00 38 FD 44 2E 10 02 A9 <A6>
0B48:47 18 7D 54 2E C9 48 90 <31>
0B50:02 A9 00 99 D8 00 88 10 <08>
0B58:AD 60 00 00 00 00 00 00 <DC>
0B60:00 00 20 20 20 00 00 00 <F9>
0B68:00 00 00 00 00 00 00 20 <9E>
0B70:20 20 00 00 00 00 00 00 <9E>
0B78:00 00 00 00 00 00 00 00 <8E>
0B80:01 00 00 00 01 00 00 00 <1F>
0B88:01 00 00 00 00 00 00 01 <20>
0B90:00 00 00 01 00 00 00 01 <B7>
0B98:00 00 48 8A 48 A5 DE D0 <B0>
```

»Paver« für zwei Spieler (Fortsetzung)

```
0BA0:06 20 7E 2E 4C 78 2E C9 <99>
0BA8:01 D0 03 20 8B 2E E6 DE <4C>
0BB0:68 AA 68 40 A5 D4 4A 4A <C9>
0BB8:4A 4A 4A 49 07 8D 04 D4 <5E>
0BC0:60 A5 D6 4A 4A 4A 4A 4A <A1>
0BC8:49 07 8D 04 D4 60 A9 64 <8C>
0BD0:8D 00 02 A9 2E 8D 01 02 <9B>
0BD8:A9 00 85 DE A9 C0 8D 0E <47>
0BE0:D4 60 00 00 A0 01 B6 DA <49>
0BE8:98 48 0A A8 B9 E3 00 D0 <C1>
0BF0:05 68 A8 4C 22 2F B9 E2 <9F>
0BF8:00 18 7D 34 2E 99 E2 00 <7A>
0C00:B9 E3 00 69 00 99 E3 00 <60>
0C08:B9 E2 00 38 FD 24 2E 99 <66>
0C10:E2 00 B9 E3 00 E9 00 99 <AB>
0C18:E3 00 68 A8 48 B9 E8 00 <1E>
0C20:AA 98 0A A8 B9 E2 00 18 <64>
0C28:7D 36 2F 99 E2 00 B9 E3 <2F>
0C30:00 69 00 99 E3 00 B9 E2 <A6>
0C38:00 38 FD 26 2F 99 E2 00 <F1>
0C40:B9 E3 00 E9 00 99 E3 00 <24>
0C48:68 A8 B9 E6 00 38 FD 46 <D0>
0C50:2F 18 7D 56 2F 99 E6 00 <86>
0C58:88 10 8B 60 00 00 00 00 <8F>
0C60:00 00 00 00 00 80 80 80 <FB>
0C68:00 00 00 00 00 00 00 00 <80>
0C70:00 80 80 80 00 00 00 00 <B0>
0C78:00 00 00 00 00 00 00 00 <90>
0C80:00 00 04 00 00 00 04 00 <21>
0C88:00 00 04 00 00 00 00 00 <21>
0C90:00 04 00 00 00 04 00 00 <B9>
0C98:00 04 00 00 A2 01 A0 02 <0C>
0CA0:B9 E3 00 D0 78 BD 84 02 <8A>
0CA8:D0 73 B5 EC D0 6F B9 D4 <3C>
0CB0:00 18 69 60 99 E2 00 B9 <FF>
0CB8:D5 00 69 00 99 E3 00 B5 <55>
0CC0:D8 18 69 04 95 E6 B5 EA <D1>
0CC8:95 E8 E0 01 F0 29 48 8A <6F>
0CD0:48 98 48 A2 00 A9 03 A0 <1E>
0CD8:00 20 E9 23 68 A8 68 AA <A4>
0CE0:68 48 8A 48 98 48 A2 02 <89>
0CE8:A9 06 A0 00 20 E9 23 68 <9D>
0CF0:A8 68 AA 68 4C E7 2F 48 <C6>
0CF8:8A 48 98 48 A2 01 A9 03 <66>
0D00:A0 00 20 E9 23 68 A8 68 <6F>
0D08:AA 68 48 8A 48 98 48 A2 <C8>
0D10:03 A9 06 A0 00 20 E9 23 <05>
0D18:68 A8 68 AA 68 88 88 CA <2A>
0D20:30 03 4C 6A 2F 60 20 FD <6B>
0D28:2F 20 46 30 20 88 30 20 <4D>
0D30:CA 30 60 A9 11 85 A0 A5 <32>
0D38:E6 C9 50 B0 36 69 20 85 <C5>
0D40:A8 A5 E2 38 E5 D4 8D AC <52>
0D48:2E A5 E3 E5 D5 8D AD 2E <D3>
0D50:C9 0A 90 04 C9 F5 90 1B <D7>
0D58:0E AC 2E 2E AD 2E 0E AC <F3>
0D60:2E 2E AD 2E 0E AC 2E 2E <BE>
0D68:AD 2E AD AD 2E 18 69 7F <87>
0D70:85 A4 60 A9 00 85 A4 85 <DF>
0D78:A6 85 E3 60 A9 12 85 A1 <A9>
0D80:A5 E7 18 69 20 85 A9 A5 <BE>
0D88:E4 38 E5 D4 8D AC 2E A5 <B3>
0D90:E5 E5 D5 8D AD 2E C9 0A <8B>
0D98:90 04 C9 F5 90 1B 0E AC <15>
0DA0:2E 2E AD 2E 0E AC 2E 2E <7F>
0DA8:AD 2E 0E AC 2E 2E AD 2E <D2>
0DB0:AD AD 2E 18 69 7F 85 A5 <88>
0DB8:60 A9 00 85 A5 60 A9 13 <40>
0DC0:85 A2 A5 E6 18 69 90 85 <A3>
0DC8:AA A5 E2 38 E5 D6 8D AC <FC>
0DD0:2E A5 E3 E5 D7 8D AD 2E <0B>
0DD8:C9 0A 90 04 C9 F5 90 1B <62>
0DE0:0E AC 2E 2E AD 2E 0E AC <3B>
0DE8:2E 2E AD 2E 0E AC 2E 2E <E6>
0DF0:AD 2E AD AD 2E 18 69 7F <00>
0DF8:85 A6 60 A9 00 85 A6 60 <B6>
0E00:A9 14 85 A3 A5 E7 C9 50 <5B>
0E08:B0 36 69 90 85 AB A5 E4 <00>
0E10:38 E5 D6 8D AC 2E A5 E5 <69>
0E18:E5 D7 8D AD 2E C9 0A 90 <F1>
0E20:04 C9 F5 90 1B 0E AC 2E <9E>
0E28:2E AD 2E 0E AC 2E 2E AD <4C>
0E30:2E 0E AC 2E 2E AD 2E AD <88>
0E38:AD 2E 18 69 7F 85 A7 60 <83>
0E40:A9 00 85 A5 85 A7 85 E5 <52>
0E48:60 00 00 01 07 07 08 07 <8F>
0E50:07 07 08 07 07 07 08 07 <8E>
0E58:07 07 08 07 01 01 01 01 <3A>
0E60:00 8F 8F 8F 8F 8F 8F 8F <6D>
0E68:8F 3F 3F 3F 3F 3F 3F 3F <84>
0E70:3F 13 31 65 31 6D 31 75 <80>
0E78:31 7E 31 86 31 8E 31 96 <D7>
0E80:31 9F 31 65 31 6D 31 75 <4B>
0E88:31 7E 31 86 31 8E 31 96 <C7>
0E90:31 9F 31 A7 31 A9 31 AB <05>
0E98:31 AD 31 08 08 1C 1C 3E <82>
```

```
0EA0:7F 7F 00 07 1F 7E 3E 0E <09>
0EA8:0C 04 00 60 60 78 7E 7F <22>
0EB0:7E 78 60 60 0C 0C 1E 3E <C6>
0EB8:7E 3F 07 00 FE FE 7C 7C <A1>
0EC0:38 38 10 10 18 18 1C 3E <61>
0EC8:3F 7E 78 60 03 03 0F 3F <39>
0ED0:7F 3F 0F 03 03 60 78 7E <14>
0ED8:3F 3E 1C 18 18 03 03 0C <87>
0EE0:0C 30 30 C0 C0 20 B6 31 <B3>
0EE8:20 0A 32 60 A5 C4 10 08 <37>
0EF0:20 F9 33 90 03 4C B3 34 <8C>
0EF8:20 5E 32 20 75 33 90 33 <AC>
0F00:20 82 32 A9 00 85 E1 AD <F0>
0F08:74 33 20 25 35 48 8A 48 <AB>
0F10:98 48 A2 00 A9 01 A0 01 <12>
0F18:20 E9 23 68 A8 68 AA 68 <07>
0F20:48 8A 48 98 48 A2 02 A9 <B5>
0F28:04 A0 01 20 E9 23 68 A8 <DF>
0F30:68 AA 68 A5 EC F0 08 A5 <D8>
0F38:EE 29 3F D0 02 C6 EC 60 <DF>
0F40:A5 C4 10 08 20 F9 33 90 <FF>
0F48:03 4C B3 34 20 70 32 20 <DB>
0F50:75 33 90 33 20 82 32 A9 <AD>
0F58:01 85 E1 AD 74 33 20 25 <DC>
0F60:35 48 8A 48 98 48 A2 01 <9C>
0F68:A9 02 A0 01 20 E9 23 68 <2A>
0F70:A8 68 AA 68 48 8A 48 98 <49>
0F78:48 A2 03 A9 05 A0 01 20 <E6>
0F80:E9 23 68 A8 68 AA 68 A5 <A2>
0F88:ED F0 08 A5 EE 29 3F D0 <C2>
0F90:02 C6 ED 60 A5 D8 4A 4A <37>
0F98:4A 18 69 80 85 E0 A5 D5 <CA>
0FA0:18 69 0B 85 DF 60 A5 D9 <79>
0FA8:4A 4A 4A 18 69 80 85 E0 <61>
0FB0:A5 D7 18 69 0B 85 DF 60 <EC>
0FB8:A5 C4 09 40 A0 00 91 DF <3E>
0FC0:A9 FF 85 C4 60 A6 EA BD <F2>
0FC8:52 33 85 84 85 86 A5 EC <98>
0FD0:F0 0A 29 01 D0 06 A9 00 <86>
0FD8:85 84 85 86 A9 7F 85 88 <D9>
0FE0:A5 D8 48 18 69 20 85 8C <32>
0FE8:68 18 69 90 85 8E A6 EB <B8>
0FF0:BD 62 33 85 87 85 85 A5 <EE>
0FF8:ED F0 0A 29 01 D0 06 A9 <5C>
1000:00 85 85 85 87 A9 7F 85 <AB>
1008:8B A5 D9 48 18 69 90 85 <BD>
1010:8F 68 18 69 20 85 8D A5 <42>
1018:D6 38 E5 D4 8D 72 33 A5 <48>
1020:D7 E5 D5 8D 73 33 C9 0A <6C>
1028:90 04 C9 F4 90 1D 0E 72 <F0>
1030:33 2E 73 33 0E 72 33 2E <C1>
1038:73 33 0E 72 33 2E 73 33 <89>
1040:AD 73 33 18 69 7F 85 89 <14>
1048:4C 19 33 A9 00 85 89 A5 <11>
1050:D4 38 E5 D6 8D 72 33 A5 <9F>
1058:D5 E5 D7 8D 73 33 C9 0A <C9>
1060:90 04 C9 F4 90 1D 0E 72 <A9>
1068:33 2E 73 33 0E 72 33 2E <19>
1070:73 33 0E 72 33 2E 73 33 <64>
1078:AD 73 33 18 69 7F 85 8A <CC>
1080:4C 51 33 A9 00 85 8A 60 <83>
1088:03 00 00 00 00 04 02 03 <41>
1090:00 06 08 07 00 05 01 00 <B9>
1098:0B 00 00 00 00 0C 0A 0B <8D>
10A0:00 0E 10 0F 00 0D 09 00 <7D>
10A8:00 00 00 A0 00 8C 74 33 <0C>
10B0:A5 DF 48 A5 E0 48 B1 DF <80>
10B8:C9 8F D0 6B C8 B1 DF C9 <31>
10C0:8F D0 06 EE 74 33 4C 9B <5B>
10C8:33 4A 4A 45 C4 29 02 D0 <B8>
10D0:56 88 C6 DF B1 DF C9 8F <85>
10D8:D0 06 EE 74 33 4C B2 33 <C2>
10E0:0A 0A 45 C4 29 08 D0 3F <C4>
10E8:E6 DF C6 E0 B1 DF C9 8F <13>
10F0:D0 06 EE 74 33 4C CA 33 <88>
10F8:4A 4A 45 C4 29 01 D0 27 <88>
1100:E6 E0 E6 E0 B1 DF C9 8F <E9>
1108:D0 06 EE 74 33 4C E2 33 <5A>
1110:0A 0A 45 C4 29 04 D0 0F <96>
1118:AD 74 33 C9 04 B0 08 68 <FB>
1120:85 E0 68 85 DF 38 60 68 <F0>
1128:85 E0 68 85 DF 18 60 A9 <A9>
1130:00 85 DC 85 DD AD 0A D2 <F7>
1138:85 DF AD 0A D2 29 07 18 <E3>
1140:69 80 69 02 85 E0 A0 00 <CA>
1148:8C 74 33 B1 DF C9 8F D0 <97>
1150:67 C8 B1 DF C9 8F D0 06 <C3>
1158:AD 0A D2 EE 74 33 4A 4A <B3>
1160:29 02 85 C4 88 C6 DF B1 <2A>
1168:DF C9 8F D0 06 AD 0A D2 <64>
1170:EE 74 33 0A 0A 29 08 05 <B7>
1178:C4 85 C4 E6 DF C6 E0 B1 <73>
1180:DF C9 8F D0 06 AD 0A D2 <FC>
1188:EE 74 33 4A 4A 29 01 05 <A3>
1190:C4 85 C4 E6 E0 E6 E0 B1 <D4>
1198:DF C9 8F D0 06 AD 0A D2 <F4>
```

```
11A0:EE 74 33 0A 0A 29 04 05 <9F>
11A8:C4 85 C4 09 40 8D 6D 2C <0D>
11B0:AD 74 33 C9 04 B0 01 60 <3D>
11B8:A5 DF 18 69 01 85 DF A5 <FB>
11C0:E0 69 00 85 E0 A5 E0 C9 <93>
11C8:88 90 07 A9 80 18 69 02 <3F>
11D0:85 E0 A5 DC 18 69 01 85 <74>
11D8:DC A5 DD 69 00 85 DD A5 <A9>
11E0:DD C9 08 B0 03 4C 10 34 <C8>
11E8:60 68 68 68 68 4C 50 37 <C9>
11F0:A9 8F 8D 6D 2C A9 FF 85 <8E>
11F8:EC 85 ED A9 07 85 EA 85 <40>
1200:DA A9 0B 85 EB 85 DB 20 <68>
1208:10 26 A9 01 85 BC AD 1F <76>
1210:D0 C9 05 D0 12 AD F9 22 <47>
1218:49 01 8D F9 22 20 04 35 <30>
1220:AD 1F D0 C9 07 D0 F9 20 <92>
1228:04 35 AD 1F D0 C9 06 D0 <A0>
1230:DD A9 00 85 EC 85 ED 85 <49>
1238:BC 60 AD F9 22 D0 0C A9 <44>
1240:00 8D 67 2C 8D 68 2C 8D <1E>
1248:69 2C 60 A9 67 8D 67 2C <AC>
1250:A9 61 8D 68 2C A9 70 8D <C9>
1258:69 2C 60 0A 0A 0A 18 69 <59>
1260:07 A8 A2 05 A5 E1 D0 17 <15>
1268:18 BD 49 2C 79 5E 35 C9 <28>
1270:1A 90 02 E9 0A 9D 49 2C <D9>
1278:88 CA 10 ED 4C 5D 35 18 <7C>
1280:BD 50 2C 79 5E 35 C9 1A <97>
1288:90 02 E9 0A 9D 50 2C 88 <39>
1290:CA 10 ED 60 00 00 00 00 <31>
1298:00 00 04 00 00 00 00 00 <3D>
12A0:00 00 02 00 00 00 00 00 <05>
12A8:00 00 01 00 00 00 00 00 <EC>
12B0:00 00 00 05 AD 08 D0 29 <73>
12B8:02 F0 46 A9 FF 85 ED A9 <AE>
12C0:0F 85 DB A9 07 85 EB A9 <3C>
12C8:22 85 D7 A9 00 85 D6 85 <CB>
12D0:E5 A9 30 85 D9 A9 0B 85 <F4>
12D8:85 85 87 48 8A 48 98 48 <BA>
12E0:A2 01 A9 07 A0 00 20 E9 <41>
12E8:23 68 A8 68 AA 68 48 8A <2C>
12F0:48 98 48 A2 03 A9 08 A0 <8F>
12F8:00 20 E9 23 68 A8 68 AA <CE>
1300:68 AD 0B D0 29 04 F0 46 <63>
1308:A9 FF 85 EC A9 0F 85 DA <1E>
1310:A9 07 85 EA A9 22 85 D5 <47>
1318:A9 00 85 D4 85 E3 A9 18 <83>
1320:85 D8 A9 03 85 84 85 86 <CB>
1328:48 8A 48 98 48 A2 00 A9 <C1>
1330:07 A0 00 20 E9 23 68 A8 <0E>
1338:68 AA 68 48 8A 48 98 48 <12>
1340:A2 02 A9 08 A0 00 20 E9 <73>
1348:23 68 A8 68 AA 68 A9 03 <86>
1350:8D 1E D0 60 AD F9 22 D0 <71>
1358:02 18 60 AD 4B 2C C9 11 <D9>
1360:B0 05 AD 52 2C C9 11 60 <05>
1368:00 00 00 00 00 00 00 00 <8E>
1370:00 00 00 00 00 00 A9 00 <E9>
1378:8D 0E D4 8D 00 D4 60 A9 <F7>
1380:FF 8D 0E D4 60 E6 CB D0 <48>
1388:02 E6 CC 60 38 E9 01 D0 <E0>
1390:FB 60 A9 00 8D 32 36 A9 <6B>
1398:38 8D 33 36 A9 F7 8D 34 <56>
13A0:36 A9 3B 8D 35 36 60 A9 <38>
13A8:00 8D 32 36 A9 38 8D 33 <50>
13B0:36 A9 FF 8D 34 36 A9 39 <C9>
13B8:8D 35 36 60 20 40 36 AD <52>
13C0:32 36 85 CB AD 33 36 85 <CE>
13C8:CC A0 00 B1 CB 8D 3A 36 <43>
13D0:20 4F 36 B1 CB 8D 38 36 <4B>
13D8:20 4F 36 B1 CB 8D 39 36 <65>
13E0:20 4F 36 20 4F 36 20 4F <95>
13E8:36 A5 CB 8D 3B 36 A5 CC <3C>
13F0:8D 3C 36 AD 3B 36 85 CB <12>
13F8:AD 3C 36 85 CC AE 3A 36 <94>
1400:B1 CB 85 CF A0 08 8E 3F <72>
1408:36 26 CF 90 04 CA 4C DC <63>
1410:36 E8 8A 18 6D 3F 36 4A <2A>
1418:09 10 8D 01 D2 AD 36 36 <70>
1420:20 56 36 88 D0 E0 20 4F <CA>
1428:36 A5 CB CD 34 36 D0 0A <DE>
1430:A5 CC CD 35 36 D0 03 4C <D1>
1438:22 37 4C CA 36 CE 38 36 <D4>
1440:D0 B1 CE 39 36 10 AC A5 <F8>
1448:CC CD 35 36 B0 0A A5 CB <4D>
1450:CD 34 36 B0 03 4C 93 36 <62>
1458:20 49 36 60 A9 04 8D 36 <54>
1460:36 20 71 36 20 86 36 20 <E0>
1468:71 36 20 86 36 20 71 36 <7C>
1470:20 86 36 20 71 36 20 86 <FD>
1478:36 20 5C 36 20 86 36 60 <45>
1480:20 26 37 20 50 29 20 BA <0C>
1488:34 20 50 29 E6 EE 20 AF <45>
1490:31 20 1E 36 B0 F0 4C 56 <A2>
1498:37 EA 00 00 00 00 00 00 <D6>
```

```
14A0:00 00 00 00 00 00 00 00 <C8>
14A8:00 00 00 00 00 00 00 00 <D0>
14B0:00 00 00 00 00 00 00 00 <D8>
14B8:00 00 00 00 00 00 00 00 <E0>
14C0:00 00 00 00 00 00 00 00 <E8>
14C8:00 00 00 00 00 00 00 00 <F0>
14D0:00 00 00 00 00 00 00 00 <F8>
14D8:00 00 00 00 00 00 00 00 <00>
14E0:00 00 00 00 00 00 00 00 <08>
14E8:00 00 00 00 00 00 00 00 <10>
14F0:00 00 00 00 00 00 00 00 <18>
14F8:00 00 00 00 00 00 00 00 <20>
1500:00 00 00 00 00 00 00 00 <2A>
1508:00 00 00 00 00 00 00 00 <32>
1510:00 00 00 00 00 00 00 00 <3A>
1518:00 00 00 00 00 00 00 00 <42>
1520:00 00 00 00 00 00 00 00 <4A>
1528:00 00 00 00 00 00 00 00 <52>
1530:00 00 00 00 00 00 FF FF <57>
1538:08 2A FF 2F AA AA AA AA <B9>
1540:AA AA AA AA AA AA AA AA <92>
1548:AA AA AA AA AA AA AA AA <BB>
1550:AA AA AA AA AA AA AA AA <C3>
1558:AA AA AA AA AA AA AA AA <2D>
1560:AA AA AA AA AA AA AA AA <35>
1568:AA AA AA AA AA AA AA AA <1D>
1570:AA AA AA AA AA AA AA AA <25>
1578:AA AA AA AA AA AA AA AA <4D>
1580:AA AA AA AA AA AA AA AA <55>
1588:55 5A AA 95 55 55 55 4A <67>
1590:AA AA AA FD 54 2E C2 88 <9E>
1598:55 6A 95 AA DA F2 22 95 <84>
15A0:55 AD 2A BB FA AA A9 53 <AE>
15A8:55 55 29 2A AB AB 55 55 <62>
15B0:55 52 AA 55 35 AA A9 AA <61>
15B8:AB 4A AA AA AA AA AA AA <75>
15C0:AA AA AA A4 AA AA AA AA <B3>
15C8:AA AA AA 95 55 AA AA AA <41>
15D0:D6 A8 A2 AB AB 2A AA AA <BA>
15D8:B4 AA A9 5A AA AA B5 AB <A1>
15E0:0C 2A BB 54 A9 6A B2 CA <55>
15E8:A5 55 AA AA AA DA B0 A2 <C8>
15F0:AB D5 4A 55 AB 34 AA 56 <83>
15F8:6A AA AA B5 AD 08 AA F5 <38>
1600:62 A5 56 D5 2A 95 5A AA <B1>
1608:AA AB AB 0C 2A BB 68 A9 <C0>
1610:55 AD 52 95 56 AA AA AB <F9>
1618:6D 18 2A BA E5 4A 55 AB <45>
1620:52 95 55 AD 55 55 B0 C3 <1A>
1628:2B 6B 34 AA 5A AA AA 55 <C3>
1630:5A AD 55 AA 30 D5 6A CD <30>
1638:2A 96 AA AA 55 5A AD 69 <8D>
1640:AA 31 4A DA B5 2A 95 6A <57>
1648:AA A5 AA B5 AA A8 93 2B <73>
1650:6C D4 AA 55 AA A6 AA AA <AB>
1658:D5 AA 8C 55 55 AB 54 A9 <35>
1660:55 9A AA AA AB 5A A8 C5 <4E>
1668:2A B6 B5 52 A5 55 5A AA <9F>
1670:AA B5 AA 98 94 AA EB 55 <4B>
1678:2A 55 55 6A AA AB 56 A9 <65>
1680:8A 95 56 B5 52 A9 55 55 <51>
1688:AA AA AD 6A 96 29 55 5A <BC>
1690:D5 4A A9 55 5A AA AA D5 <B5>
1698:AA 62 A5 55 6A B5 2A 95 <2B>
16A0:55 5A AA AA D5 AA 62 95 <58>
16A8:55 AB 55 2A 95 55 5A AA <CE>
16B0:AA D5 AA 62 A5 55 6B 55 <FC>
16B8:2A A5 55 5A AA AA B5 6A <2E>
16C0:A6 1A 54 DA B5 52 AA 96 <12>
16C8:AA AA AA AA D5 6A 95 19 <D4>
16D0:55 5A B5 2A AA 5A AA AA <AB>
16D8:AA AA D5 5A A3 45 55 5A <42>
16E0:AB 52 AA 5A AA AA 95 6A <D0>
16E8:AA D5 55 68 AA 95 56 AA <A9>
16F0:AA AA AA AA AA 95 55 55 <5D>
16F8:6A AA AD 5A 95 52 AA AA <06>
1700:AA AA AA 95 55 55 55 6A <4C>
1708:AA AA AA AA D5 65 2A AA <BE>
1710:AA AA AA AA AA AA AA AA <8E>
1718:AA AA AA AA AA AA D5 55 <68>
1720:55 2A AA AA AA AA AA AA <28>
1728:AA AA AA AA AA AA AA AA <56>
1730:AA D5 2A AA AA AA AA AA <43>
1738:AA AA AA AA AA AA AA AA <86>
1740:AA AA AA D5 4A AA AA AA <66>
1748:AA AA AA AA AA AA AA AA <B7>
1750:AA AA AA AA AA AA D4 AA <23>
1758:AA AA AA AA AA AA AA AA <29>
1760:AA AA AA AA AA AA AA AA <21>
1768:AD 52 AA AA AA AA AA AA <AB>
1770:AA AA AA AA AA AA AA AA <31>
1778:AA AA AD 55 2A AA AA AA <56>
1780:AA AA AA AA AA AA AA AA <01>
1788:AA AA AA AB 56 52 95 5A <8F>
1790:AB 2A AA B6 AA AA 2A AD <38>
1798:4A AA AA AD 55 55 24 AA <40>
17A0:B6 AA 95 5A AA AA 55 56 <40>
17A8:AA A5 AA AA D5 56 AA 12 <07>
17B0:AD D5 2A 56 AA AA AA AA <F6>
17B8:A9 2D A9 5A AA AD 6A AA <97>
17C0:0A AF 8A AA 6B 52 AA AA <1B>
17C8:B2 AA 92 EC 95 AA B5 56 <7A>
17D0:AA 82 AB E2 AA AA D4 AA <31>
17D8:AA AA AA A9 5A A9 6A AA <95>
17E0:D5 6A A8 2A FA 25 AA AD <4F>
17E8:2A AA AA AA AA A5 6A A5 <BD>
17F0:AA AA D5 AA 90 AB E2 5A <60>
17F8:AA D4 9A AA AA AA AA 55 <EC>
1800:AA 9A AA AB 56 AA 82 AF <A8>
1808:84 DA AB 49 5A AA AA AA <53>
1810:AA 56 AA AA AA AA D6 AA <EC>
1818:84 AF A0 B6 AA D2 5A CA <56>
1820:AA AA AA 95 AA AA AA AA <11>
1828:B5 A5 A0 AB E2 56 AA D4 <0A>
1830:9A CA AA AA AA A9 6A AA <F0>
1838:AA AA AD 55 6A 0A FA 15 <27>
1840:AA AD 2A AA AA AA AA AA <8B>
1848:56 AA AA AA AB 55 55 A0 <5E>
1850:AF 89 5A AA AA AA AA AA <7B>
1858:AA AA A5 6A A9 AA AA D5 <AB>
1860:55 5A 0A F9 15 AA AA AA <AF>
1868:AA AA AA AA AA A5 AA AA <0F>
1870:AA AA D5 55 64 4D D2 5A <B2>
1878:AA AA AA AA AA AA AA AA <53>
1880:95 AA AA AA AB 55 55 52 <79>
1888:56 CA 6A AA AA AA AA AA <21>
1890:AA AA A9 6A AA AA AA B5 <DC>
1898:55 5A 45 6D 29 AA AA AA <36>
18A0:AA AA AA AA AA AA 9A AA <C5>
18A8:AA AA AB 55 55 92 5B 49 <A5>
18B0:6A AA AA AA AA AA AA AA <F5>
18B8:AA AA AA AA AA AA AA B5 <1E>
18C0:55 2A AA AA AA AA AA AA <86>
18C8:AA AA AA AA AA AA AA AA <43>
18D0:AA AA AA B5 2A AA AA AA <77>
18D8:AA AA AA AA AA AA AA AA <AD>
18E0:AA AA AA AA AA AA AA AA <A4>
18E8:AA AA AA AA AA AA AA AA <9C>
18F0:AA AA AA AA AA AA AA AA <B5>
18F8:AA AA AA AA AA AA AA AA <CD>
1900:AA AA AA AA AA AA AA AA <7A>
1908:AA AA AA AA AA AA AA AA <82>
1910:AA AA AA AA AA AA AA D5 <95>
1918:55 2A AA AA AA AA AA AA <54>
1920:AA AA AA AA AA AA AA AA <5A>
1928:AA AA AA AA AA 00 00 00 <BE>
1930:00 00 00 00 00 00 00 00 <62>
1938:00 00 00 00 00 00 FF 7E <E6>
1940:3C 18 00 00 00 00 01 03 <1B>
1948:07 0F 0F 07 03 01 FF 7F <6C>
1950:3F 1F 0F 07 03 01 00 00 <58>
1958:00 00 18 3C 7E FF FF 7E <AD>
1960:3C 18 18 3C 7E FF 01 03 <6A>
1968:07 0F 1F 3F 7F FF 7F 3F <ED>
1970:1F 0F 0F 1F 3F 7F 80 C0 <7C>
1978:E0 F0 F0 E0 C0 80 FF FE <8E>
1980:FC F8 F0 E0 C0 80 81 C3 <70>
1988:E7 FF FF E7 C3 81 FF FF <51>
1990:FF FF FF E7 C3 81 80 C0 <31>
1998:E0 F0 F8 FC FE FF FF FE <5D>
19A0:FC F8 F8 FC FE FF 81 C3 <3D>
19A8:E7 FF FF FF FF FF FF FF <CE>
19B0:FF FF FF FF FF FF 00 3F <24>
19B8:33 33 33 3F 3F 00 00 3C <61>
19C0:3C 0C 0C 3F 3F 00 00 3F <42>
19C8:03 3F 30 3F 3F 00 00 3F <3E>
19D0:03 0F 03 3F 3F 00 00 33 <C8>
19D8:33 3F 3F 03 03 00 00 3F <E2>
19E0:30 3F 03 3F 3F 00 00 3F <7F>
19E8:30 3F 33 3F 3F 00 00 3F <55>
19F0:3F 03 03 03 03 00 00 3F <6A>
19F8:33 3F 33 3F 3F 00 00 3F <26>
1A00:33 3F 03 3F 3F 00 00 00 <DB>
1A08:00 FF FF 00 00 00 00 00 <DB>
1A10:00 3F 3F 00 00 00 00 00 <FB>
1A18:00 FC FC 00 00 00 1F 7F <88>
1A20:FF FF FF FF FF FF F8 FE <45>
1A28:FF FF FF FF FF FF FF FF <5C>
1A30:FF FF FF FF FE F8 00 3C <7F>
1A38:66 6E 6E 60 3E 00 00 3F <97>
1A40:33 33 3F 33 33 00 00 FC <EA>
1A48:CC FF CF CF FF 00 00 3F <40>
1A50:33 30 30 33 3F 00 00 F0 <4D>
1A58:FC CC CF FF FC 00 00 3F <A6>
1A60:30 3C 30 3F 3F 00 00 3F <CE>
1A68:3F 30 3C 30 30 00 00 FC <00>
1A70:C0 CF C3 FF FF 00 00 33 <EB>
1A78:33 3F 3F 33 33 00 00 3F <F9>
1A80:0C 0C 0C 3F 3F 00 00 03 <20>
1A88:03 03 33 3F 3F 00 00 33 <66>
1A90:33 3C 3C 37 37 00 00 30 <52>
1A98:30 30 30 3F 3F 00 00 C3 <A7>
1AA0:FF FF FF C3 C3 00 00 C3 <01>
1AA8:F3 FF FF CF C3 00 00 3F <37>
1AB0:33 33 33 3F 3F 00 00 3F <9E>
1AB8:33 3F 3F 30 30 00 00 FC <2E>
1AC0:CC CC CC FC FF 00 00 3F <3D>
1AC8:33 33 3F 3C 33 00 00 3F <5F>
1AD0:30 3F 03 3F 3F 00 00 3F <69>
1AD8:3F 0C 0C 0C 0C 00 00 33 <84>
1AE0:33 33 33 3F 3F 00 00 33 <E1>
1AE8:33 37 3E 1C 10 00 00 C3 <10>
1AF0:F3 FF 3F 3F 33 00 00 33 <E5>
1AF8:33 1E 1E 33 33 00 00 33 <10>
1B00:33 3F 3F 0C 0C 00 00 3F <C7>
1B08:07 0C 30 3F 3F 00 00 00 <B0>
1B10:00 00 00 30 30 00 3F C0 <09>
1B18:CF CC CC CF C0 3F F0 0C <07>
1B20:CC 0C 0C CC 0C F0 FF FF <8A>
1B28:FF FF FF FF 7F 1F AA 55 <86>
1B30:AA 55 AA 55 AA 55 00 36 <B3>
1B38:7F 7F 3E 1C 08 00 18 18 <5F>
1B40:18 1F 1F 18 18 18 03 03 <55>
1B48:03 03 03 03 03 03 18 18 <7D>
1B50:18 F8 F8 00 00 00 18 18 <D7>
1B58:18 F8 F8 18 18 18 00 00 <3A>
1B60:00 F8 F8 18 18 18 03 07 <43>
1B68:0E 1C 38 70 E0 C0 C0 E0 <18>
1B70:70 38 1C 0E 07 03 01 03 <9A>
1B78:07 0F 1F 3F 7F FF 00 00 <BD>
1B80:00 00 0F 0F 0F 0F 80 C0 <FF>
1B88:E0 F0 F8 FC FE FF 0F 0F <82>
1B90:0F 0F 00 00 00 00 F0 F0 <E3>
1B98:F0 F0 00 00 00 00 FF FF <BF>
1BA0:00 00 00 00 00 00 00 00 <D6>
1BA8:00 00 00 00 FF FF 00 00 <D2>
1BB0:00 00 F0 F0 F0 F0 00 1C <3F>
1BB8:1C 77 77 08 1C 00 00 00 <CA>
1BC0:00 1F 1F 18 18 18 00 00 <25>
1BC8:00 FF FF 00 00 00 18 18 <E6>
1BD0:18 FF FF 18 18 18 00 00 <50>
1BD8:3C 7E 7E 7E 3C 00 00 00 <15>
1BE0:00 00 FF FF FF FF C0 C0 <19>
1BE8:C0 C0 C0 C0 C0 C0 00 00 <63>
1BF0:00 FF FF 18 18 18 18 18 <A8>
1BF8:18 FF FF 00 00 00 F0 F0 <A9>
1C00:F0 F0 F0 F0 F0 F0 18 18 <B0>
1C08:18 1F 1F 00 00 00 78 60 <46>
1C10:78 60 7E 18 1E 00 00 18 <56>
1C18:3C 7E 18 18 18 00 00 18 <2B>
1C20:18 18 7E 3C 18 00 00 18 <D6>
1C28:30 7E 30 18 00 00 00 18 <77>
1C30:0C 7E 0C 18 00 00 00 18 <D8>
1C38:3C 7E 7E 3C 18 00 00 00 <F1>
1C40:3F 03 3F 33 3F 00 00 30 <7D>
1C48:30 3F 33 3F 3F 00 00 00 <AC>
1C50:3F 30 30 3F 3F 00 00 03 <23>
1C58:03 3F 33 3F 3F 00 00 00 <36>
1C60:3F 33 3F 30 3F 00 00 0F <B0>
1C68:0F 0C 3F 0C 0C 00 00 00 <34>
1C70:00 3F 33 3F 03 3F 00 30 <07>
1C78:30 3F 3F 33 33 00 3C 3C <E1>
1C80:00 3C 3C 3C 3C 00 03 03 <AD>
1C88:00 03 03 03 1F 1F 00 30 <B7>
1C90:30 33 3C 33 33 00 00 3C <3E>
1C98:0C 0C 0C 3C 3F 00 00 00 <11>
1CA0:33 3F 3F 33 33 00 00 00 <B6>
1CA8:3F 3F 33 33 33 00 00 00 <03>
1CB0:3F 33 33 3F 3F 00 00 00 <21>
1CB8:3F 33 3F 3F 30 30 00 00 <DA>
1CC0:3F 33 3F 3F 03 03 00 00 <D4>
1CC8:3F 3F 33 30 30 00 00 00 <5A>
1CD0:3F 30 3F 03 3F 00 00 0C <D1>
1CD8:3F 0C 0C 0F 0F 00 00 00 <9D>
1CE0:33 33 33 3F 3F 00 00 00 <B2>
1CE8:33 33 33 3F 0C 00 00 00 <F0>
1CF0:C3 F3 FF FF C3 00 00 00 <2C>
1CF8:33 3F 0C 3F 33 00 00 00 <A0>
1D00:33 33 33 3F 0C 3C 00 00 <48>
1D08:3F 03 0C 30 3F 00 00 18 <31>
1D10:3C 7E 7E 18 3C 00 18 18 <43>
1D18:18 18 18 18 18 18 00 7E <88>
1D20:78 7C 6E 66 06 00 08 18 <A1>
1D28:38 78 38 18 08 00 10 18 <3D>
1D30:1C 1E 1C 18 10 9B 00 00 <D3>
1D38:00 00 00 00 00 00 00 00 <72>
1D40:00 00 00 00 00 00 00 00 <7A>
1D48:00 00 00 00 00 00 00 00 <82>
1D50:00 00 00 00 00 00 00 00 <8A>
1D58:00 00 00 00 00 00 00 00 <92>
1D60:00 00 00 00 00 00 00 00 <9A>
1D68:00 00 00 00 00 00 00 00 <A2>
1D70:00 00 00 00 00 00 00 00 <AA>
1D78:00 00 00 00 00 00 00 00 <B2>
1D80:00 00 00 00 00 00 E0 02 <7C>
1D88:E1 02 DD 22 00 00 00 00 <23>
```

»Paver« (Schluß)

Die Horror-Höhle

Steuern Sie einen wagemutigen Forscher durch ein höhlenartiges Labyrinth. Doch Vorsicht: Viele gemeine Fallen machen die Höhle zur Hölle.

Die Zeiten, in denen Bergsteiger und Höhlenforscher noch mit Seilen herumgeklettert sind, sind schon lange vorbei. Statt dessen verwendet man heutzutage Hubschrauber-Rucksäcke, mit denen die Forscher durch die Luft fliegen, statt umständlich zu klettern. Diese Technik macht das Erforschen von Höhlen zwar etwas schneller, aber nicht unbedingt leichter, besonders wenn es einen in ein so gefährliches Höhlensystem verschlagen hat wie den Helden unseres Spieles.

Auf den Heliman warten Energiesperren und fliegende Felsbrocken. Die Wände darf der Spieler ebenfalls nicht berühren. Die letzte große Schwierigkeit sind unterirdische Seen, die den fliegenden Forscher für immer auf den Grund zu ziehen drohen. Wie Sie sehen, muß man auf allerlei achten, um lebend zum Ausgang zu gelangen.

Um Ihnen die Aufgabe etwas zu erleichtern, gibt es in den Höhlen kleine Fackeln, die Sie schadlos berühren dürfen. Die Fackeln haben sogar eine sehr positive Wirkung. Wenn Sie eines ihrer zehn Leben verlieren, beginnen Sie das Spiel wieder an der letzten Fackel.

Die zweite Erleichterung ist der Turbospeed für den Rotor. Durch Drücken des Feuerknopfs verdoppelt sich die Fluggeschwindigkeit, doch leider auch der Treibstoffverbrauch. Und 1000 Liter sind sehr schnell weg.

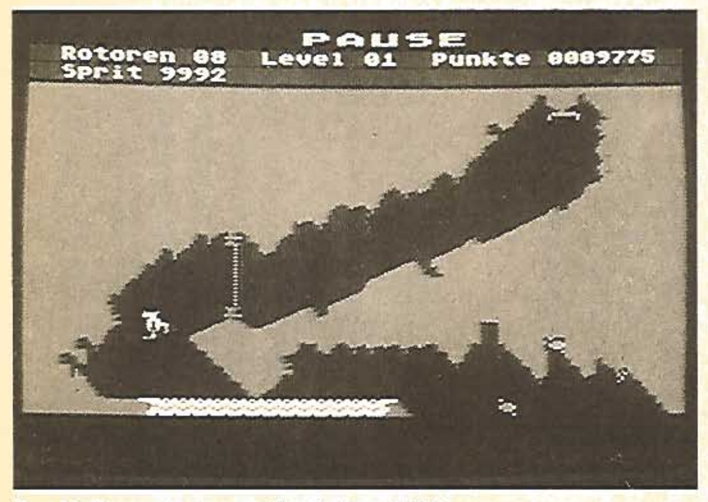
Um eine Pause einzulegen, brauchen Sie nur eine beliebige Taste zu drücken. Das Spiel stoppt solange, bis Sie eine andere Taste drücken oder weiterfliegen.

Gesteuert wird Heliman mit dem Joystick in Port 1. Wenn Sie das Ziel in der Höhle erreicht haben, bekommen Sie für den verbleibenden Treibstoff Punkte und der nächste Schwierigkeitsgrad wird von Diskette geladen. Ein neues Leben erhalten Sie für 10000 Punkte.

Wenn Sie die die abgedruckten Level sicher beherrschen oder wenn Ihnen diese zu schwer sind, können Sie den Editor benutzen, mit dem Sie eigene Schwierigkeitsstufen ganz nach Ihrem Wunsch entwerfen. Um den Editor zu starten, drücken Sie einfach die <SELECT>-Taste. Ein Zeichen wählen Sie aus, indem Sie mit dem Pfeil zum gewünschten Symbol auf der Auswahlleiste am unteren Bildrand fahren und es dann mit dem Feuerknopf aktivieren. Das Zeichen setzen Sie dann wiederum durch Druck auf den Feuerknopf in das Spielfeld. Wenn Sie mit dem Pfeil an den oberen oder unteren Bildrand stoßen, scrollt das Bild automatisch in diese Richtung.

Im Editor liegen einige Funktionen auch auf der Tastatur. Mit <L> laden Sie ein Bild und mit <S> speichern Sie es. Unter welchem Namen es geladen oder gespeichert wird, bestimmt die Bildnummer. Um diese zu verändern, drücken Sie einfach <B>. Dadurch können Sie auch einen bestehen Level verändern und unter anderem Namen speichern. Die Bildnummer wird beim Editor in der oberen Zeile eingeblendet. Mit <RETURN> können Sie zu einem Testflug starten, ein weiterer Druck auf die <RETURN>-Taste beendet den kleinen Rundflug. Mit der <START>-Taste verlassen Sie den Editor und beginnen ein neues Spiel.

Wenn Sie Heliman mit DOS 2.5 laden, geben Sie als Startadresse »A800« ein. (Klaus Fecker/gn/hf)



Der Heliman in der gefährlichen Höhle

Steckbrief	
Programm:	Heliman
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Diskette

```
Programmname :HELIMANN.COM
Länge        :4328 Bytes
-------------------------------------

0000:FF FF 00 A8 DD B8 4C 30 <C8>
0008:A8 4C 13 A8 44 3A 42 49 <34>
0010:4C 44 30 30 2E 44 41 54 <59>
0018:9B A2 10 9D 4A 03 A9 00 <DB>
0020:9D 4B 03 A9 03 9D 42 03 <92>
0028:A9 06 9D 44 03 A9 A8 9D <FD>
0030:45 03 20 56 E4 60 A9 00 <EE>
0038:8D 8D 04 8D 96 04 8D F0 <43>
0040:6E 8D F1 6E 8D F2 6E 8D <37>
0048:F3 6E 8D C0 02 8D C1 02 <B0>
0050:8D C4 02 8D C5 02 8D C6 <A7>
0058:02 8D C7 02 AD 42 03 8D <B3>
0060:AE 04 AD 43 03 8D AF 04 <9F>
0068:AD 44 03 8D B0 04 AD 45 <EE>
0070:03 8D B1 04 AD 48 03 8D <84>
0078:B2 04 AD 49 03 8D B3 04 <34>
0080:AD 4A 03 8D B4 04 AD 4B <65>
0088:03 8D B5 04 20 0F B2 A9 <8D>
0090:0C 85 D8 A9 A8 85 D9 A0 <2E>
0098:00 A9 30 91 D8 C8 91 D8 <AC>
00A0:A9 13 85 DC A9 B1 85 DD <80>
00A8:A9 00 8D B9 04 A9 50 8D <8A>
00B0:BA 04 20 B5 A8 20 FD B3 <58>
00B8:4C 77 A9 AC B9 04 B9 D6 <BB>
00C0:A8 91 DC C8 CC BA 04 D0 <16>
00C8:F5 60 AD 1F D0 C9 06 F0 <80>
00D0:05 C9 05 F0 04 60 4C 30 <8F>
00D8:A8 4C BB B1 80 80 B2 EF <78>
00E0:F4 EF F2 E5 EE 80 91 90 <43>
00E8:80 80 AC E5 F6 E5 EC 80 <0B>
00F0:90 90 80 80 B0 F5 EE EB <9C>
00F8:F4 E5 80 90 90 90 90 90 <A3>
0100:90 90 80 80 80 80 B3 F0 <C7>
0108:F2 E9 F4 80 99 99 99 99 <E9>
0110:80 80 80 80 80 80 80 80 <ED>
0118:80 80 80 80 80 80 80 80 <E5>
0120:80 80 80 80 80 80 80 80 <DD>
0128:80 80 80 80 80 80 80 80 <D5>
0130:80 80 80 80 80 80 80 8A <D7>
0138:8A 8A 80 80 B4 E5 F3 F4 <0D>
0140:E6 EC F5 E7 80 80 8A 8A <D8>
0148:8A 80 80 80 80 80 80 80 <B0>
0150:80 80 80 80 80 80 80 80 <AD>
0158:80 80 80 80 80 80 80 80 <A5>
0160:80 80 80 80 80 80 80 80 <9D>
0168:80 80 80 80 80 80 80 80 <95>
0170:80 80 80 80 80 80 80 80 <8D>
0178:80 80 80 80 80 20 3D B2 <B3>
0180:A9 AC 8D A2 04 A9 55 8D <18>
0188:A3 04 A9 50 8D 07 D4 A9 <48>
0190:3A 8D 2F 02 A9 02 8D 1D <66>
0198:D0 A9 36 8D C0 02 A9 CC <B1>
01A0:8D C1 02 A9 C8 8D C4 02 <5E>
01A8:A9 74 8D C7 02 A9 52 8D <3E>
01B0:C5 02 A9 0A 8D C6 02 A9 <7B>
01B8:20 8D 6F 02 A9 EB 85 58 <8E>
01C0:A9 B7 85 59 A9 C0 8D 0E <7A>
01C8:D4 A9 08 8D 97 04 A9 00 <C6>
01D0:8D 92 04 8D 86 04 8D B7 <13>
01D8:04 8D B6 04 8D 99 04 A9 <8E>
01E0:FF 8D FC 02 A0 00 B9 2D <C2>
01E8:B0 99 20 06 C8 C0 32 D0 <FF>
01F0:F5 A9 20 8D 30 02 A9 06 <45>
01F8:8D 31 02 AD 2D 06 8D A6 <DB>
0200:04 8D B8 04 AD 2E 06 8D <5F>
0208:A7 04 18 AD 23 06 8D A9 <B2>
0210:04 69 0E 8D B7 04 8D AA <95>
0218:04 AD 24 06 8D AC 04 69 <B2>
0220:00 8D AB 04 8D 24 06 18 <15>
0228:AD B7 04 69 9E 8D 95 04 <D9>
0230:AD 24 06 69 00 8D 9D 04 <5A>
```

```
0238:AD 8D 04 C9 01 D0 0A A9 <07>
0240:26 8D 27 06 A9 A9 8D 28 <2D>
0248:06 A9 80 8D 00 02 A9 B2 <7F>
0250:8D 01 02 A9 77 8D A0 04 <D2>
0258:A9 88 8D A1 04 A9 66 8D <5A>
0260:A4 04 20 77 AE A9 68 8D <14>
0268:01 D2 4C 4A AB AD 8D 04 <B8>
0270:C9 01 D0 01 60 18 B1 DC <C5>
0278:69 01 91 DC 4C 88 AA AD <41>
0280:8D 04 C9 01 D0 01 60 18 <CE>
0288:B1 DC E9 00 91 DC A0 00 <B4>
0290:B1 DC C9 8F F0 0A C9 9A <5B>
0298:F0 19 C8 C0 50 D0 F1 60 <DB>
02A0:A9 99 91 DC 88 18 B1 DC <57>
02A8:E9 00 C9 7F F0 14 91 DC <11>
02B0:4C 8A AA A9 90 91 DC 88 <4D>
02B8:18 B1 DC 69 01 91 DC 4C <95>
02C0:8A AA AD 8E 04 C9 01 D0 <B7>
02C8:06 A9 00 8D 8E 04 60 4C <92>
02D0:30 A8 EE 92 04 A9 11 8D <2F>
02D8:A0 04 A9 22 8D A1 04 A9 <80>
02E0:00 8D A4 04 20 10 AC 20 <97>
02E8:92 AD A9 33 8D A0 04 A9 <E5>
02F0:44 8D A1 04 A9 22 8D A4 <D7>
02F8:04 20 10 AC 20 92 AD AD <D5>
0300:92 04 C9 02 D0 CC A9 00 <8C>
0308:8D 92 04 AD 78 02 C9 07 <21>
0310:F0 35 C9 0E F0 25 C9 0D <07>
0318:F0 27 C9 0B F0 2F C9 06 <7D>
0320:F0 10 C9 05 F0 03 4C CC <9B>
0328:AA EE 81 04 EE 80 04 4C <D0>
0330:CC AA CE 81 04 EE 80 04 <35>
0338:4C CC AA CE 81 04 4C CC <E8>
0340:AA EE 81 04 4C CC AA EE <B3>
0348:80 04 4C CC AA 4C AD AB <D8>
0350:EE 92 04 A9 77 8D A0 04 <3A>
0358:A9 88 8D A1 04 A9 66 8D <50>
0360:A4 04 20 10 AC 20 92 AD <EA>
0368:A9 55 8D A0 04 A9 66 8D <C0>
0370:A1 04 A9 44 8D A4 04 20 <54>
0378:10 AC 20 92 AD AD 92 04 <9F>
0380:C9 02 D0 CC A9 00 8D 92 <9B>
0388:04 AD 78 02 C9 07 F0 29 <76>
0390:C9 0B F0 1F C9 0E F0 0F <9B>
0398:C9 0D F0 11 C9 0A F0 1C <BF>
03A0:C9 09 F0 21 4C 4A AB CE <08>
03A8:81 04 4C 4A AB EE 81 04 <B0>
03B0:4C 4A AB CE 80 04 4C 4A <96>
03B8:AB 4C 41 AB CE 81 04 CE <44>
03C0:80 04 4C 4A AB EE 81 04 <C9>
03C8:CE 80 04 4C 4A AB CE 81 <F8>
03D0:04 20 6C AC 4C 23 AC A0 <0D>
03D8:32 20 79 AA AD 80 04 8D <AA>
03E0:A2 04 AD 81 04 8D A3 04 <F3>
03E8:AD 2D 06 8D A6 04 AD 2E <F6>
03F0:06 8D A7 04 AD A5 04 8D <C7>
03F8:B6 04 4C 23 AC EE 81 04 <90>
0400:4C 23 AC 20 6D AE 4C 23 <44>
0408:AC AD 92 04 C9 01 D0 19 <08>
0410:EE 81 04 4C 23 AC AD 04 <CB>
0418:D0 C9 04 F0 BA C9 02 F0 <E6>
0420:E2 C9 08 F0 E4 C9 01 F0 <7F>
0428:DA 20 C4 A8 AE 81 04 E0 <9B>
0430:BE F0 98 E0 46 F0 C6 AD <90>
0438:80 04 8D 00 D0 8D 01 D0 <39>
0440:AC A4 04 B9 35 AF 9D F6 <AA>
0448:53 E8 C8 CC A0 04 D0 F3 <51>
0450:E8 9D F6 53 AE 81 04 B9 <DC>
0458:35 AF 9D F6 54 E8 C8 CC <5B>
0460:A1 04 D0 F3 20 20 B5 EE <B9>
0468:AD 04 AD AD 04 C9 80 F0 <7E>
0470:01 60 A0 21 B1 DC 8D 8F <77>
0478:04 A0 25 20 67 AA A0 21 <87>
0480:B1 DC CD 8F 04 F0 05 A0 <74>
0488:0B 20 67 AA A0 33 20 79 <08>
0490:AA A9 00 8D AD 04 AD 2E <E7>
0498:06 C9 6B D0 48 AD 2D 06 <75>
04A0:C9 A8 D0 41 A9 00 8D 43 <E8>
04A8:06 AD 81 04 C9 BD D0 73 <7B>
04B0:AD 8D 04 C9 01 F0 69 A9 <D1>
04B8:01 8D 8E 04 A0 32 20 79 <31>
04C0:AA AD 8E 04 C9 01 D0 20 <E4>
04C8:A0 21 B1 DC 8D 8F 04 A0 <27>
04D0:24 20 67 AA A0 21 B1 DC <3F>
04D8:CD 8F 04 F0 05 A0 0B 20 <87>
04E0:67 AA 4C B6 AC 4C 1E AD <F3>
04E8:A9 00 8D F0 6E 8D 8E 04 <B6>
04F0:8D F1 6E 8D F2 6E 8D F3 <47>
04F8:6E 8D C0 02 8D C1 02 8D <72>
0500:C5 02 8D C6 02 A0 15 20 <54>
0508:67 AA A9 2D 8D B9 04 A9 <5C>
0510:34 8D BA 04 20 B5 A8 20 <13>
0518:16 B4 20 FD B3 4C 77 A9 <48>
0520:4C BB B1 60 EE A5 04 AD <12>
0528:A5 04 C9 08 F0 01 60 A2 <25>
0530:00 18 AD 2D 06 69 28 8D <3B>
0538:B8 04 8E A5 04 AD 2E 06 <5E>
0540:69 00 8D 2E 06 4C 34 B5 <EE>
0548:AD 24 06 CD 9D 04 D0 1C <01>
0550:AD AA 04 8D B7 04 AD AB <30>
0558:04 8D 24 06 20 CD AD A9 <A4>
0560:01 8D A8 04 AD 23 06 CD <0E>
0568:95 04 F0 DC CE A8 04 AD <43>
0570:A8 04 C9 00 F0 01 60 A2 <54>
0578:08 18 AD 23 06 69 01 8D <B9>
0580:B7 04 8E A8 04 AD 24 06 <D4>
0588:69 00 8D 24 06 60 AD 8D <CF>
0590:04 C9 01 D0 03 4C BB B1 <5B>
0598:A2 00 8E 1E D0 86 4D A0 <56>
05A0:FE 88 C0 00 D0 FB E8 E0 <9D>
05A8:03 D0 F4 AD FC 02 C9 0C <F6>
05B0:F0 DC C9 FF D0 34 A2 78 <BF>
05B8:AD 84 02 C9 00 F0 04 8E <5C>
05C0:00 D2 60 A9 02 8D 92 04 <61>
05C8:A9 6E 8D 00 D2 A0 33 20 <C7>
05D0:79 AA 60 20 C4 A8 A2 00 <91>
05D8:8E 01 D2 86 4D A0 00 C8 <3D>
05E0:C0 32 D0 FB E8 E0 34 D0 <FA>
05E8:F4 60 A9 FF 8D FC 02 20 <6A>
05F0:CD AD 20 CD AD EE A8 04 <DF>
05F8:AD A8 04 C9 08 F0 46 A2 <08>
0600:00 20 C4 A8 AD 78 02 85 <D7>
0608:4D C9 0F D0 1C AD FC 02 <5D>
0610:C9 0C F0 0B C9 FF D0 11 <25>
0618:E0 01 D0 D3 4C FB AD AD <D0>
0620:8D 04 C9 01 D0 DB 4C BB <5A>
0628:B1 A9 FF 8D FC 02 AD AA <9A>
0630:04 8D B7 04 AD AB 04 8D <45>
0638:24 06 A9 00 8D A8 04 A9 <A4>
0640:68 8D 01 D2 60 A2 00 18 <42>
0648:AD 23 06 E9 00 8D B7 04 <66>
0650:8E A8 04 AD 24 06 E9 00 <63>
0658:8D 24 06 CD AC 04 F0 03 <99>
0660:4C E9 AD AD 23 06 CD A9 <30>
0668:04 D0 F5 A2 01 8E A8 04 <B4>
0670:4C FB AD A9 04 8D 43 06 <03>
0678:A0 0B 20 79 AA AD A6 04 <76>
0680:8D B8 04 AD A7 04 8D 2E <28>
0688:06 AD B6 04 8D A5 04 A9 <89>
0690:00 A2 00 9D F6 53 9D F6 <0B>
0698:54 E8 E0 FF D0 F5 AD A2 <02>
06A0:04 8D 80 04 18 AD A3 04 <BE>
06A8:8D 81 04 8D 8B 04 69 11 <2B>
06B0:8D 8C 04 A9 E8 8D 03 D2 <7E>
06B8:20 92 AD 20 20 B5 20 92 <A7>
06C0:AD 20 20 B5 20 92 AD 20 <3B>
06C8:20 B5 20 23 AC 20 92 AD <00>
06D0:20 20 B5 20 92 AD 20 20 <2D>
06D8:B5 20 92 AD AE 8B 04 A9 <70>
06E0:FF 9D F6 53 9D F6 54 EE <DA>
06E8:02 D2 EE 02 D2 EE 8B 04 <B1>
06F0:EC 8C 04 D0 C3 A9 00 8D <99>
06F8:03 D2 9D F6 53 9D F6 54 <7A>
0700:AD 96 04 C9 00 F0 04 20 <3F>
0708:E4 AD 60 A9 01 8D 96 04 <75>
0710:8D A8 04 20 5E AD 20 CD <F8>
0718:AD 20 CD AD AD 78 02 C9 <61>
0720:0F D0 03 4C 0D AF AD AA <F5>
0728:04 8D B7 04 8D 23 06 AD <24>
0730:AB 04 8D 24 06 A9 00 8D <CC>
0738:A8 04 60 00 00 3C 18 9C <C2>
0740:90 80 80 C0 00 28 38 10 <43>
0748:10 10 1C 00 00 00 00 1C <01>
0750:1C 10 38 28 68 28 00 00 <DD>
0758:00 00 10 00 00 00 00 3C <A4>
0760:18 1C 10 00 00 C0 80 A8 <EB>
0768:B8 90 10 10 70 00 00 00 <FC>
0770:00 1C 1C 10 38 28 68 28 <20>
0778:00 00 00 00 10 00 00 00 <07>
0780:00 3C 18 38 08 00 00 03 <47>
0788:01 15 1D 09 08 08 38 00 <51>
0790:00 00 00 38 38 08 1C 14 <3E>
0798:16 14 00 00 00 00 08 00 <C6>
07A0:00 00 00 3C 18 39 09 01 <17>
07A8:01 03 00 14 1C 08 08 08 <52>
07B0:0E 00 00 00 00 38 38 08 <1F>
07B8:1C 14 16 14 00 00 00 00 <DD>
07C0:08 00 00 FF 3F CC F3 FF <D4>
07C8:3F CC F3 AA 28 55 28 28 <6D>
07D0:28 28 AA 28 28 E3 0C C0 <59>
07D8:33 00 30 F3 FF 3F CC F3 <B6>
07E0:FF 3F CC AA 28 28 55 28 <02>
07E8:28 28 AA 28 28 0B CC 0C <5D>
07F0:C0 30 0C CC F3 FF 3F CC <07>
07F8:F3 FF 3F AA 28 28 28 55 <A5>
0800:28 28 AA 28 28 2C 33 30 <6C>
0808:0C 0C 30 3F CC F3 FF 3F <82>
0810:CC F3 FF AA 28 28 28 28 <A7>
0818:55 28 AA 28 28 08 C0 0C <81>
0820:F3 00 0C 01 00 14 55 55 <89>
0828:14 00 40 40 00 14 55 55 <9E>
0830:14 00 01 70 70 56 54 B0 <96>
0838:10 42 13 B1 00 82 10 64 <E8>
0840:00 64 24 24 24 24 24 24 <3C>
0848:24 24 24 24 24 24 24 24 <56>
0850:24 24 24 24 24 24 04 41 <2C>
0858:20 06 00 00 00 00 00 00 <F9>
0860:00 00 00 30 21 35 33 25 <DC>
0868:00 00 00 00 00 00 00 00 <78>
0870:68 65 6C 6C 69 6D 61 6E <43>
0878:00 00 00 00 00 80 A9 B3 <90>
0880:B4 80 B6 AF AE 80 EB EC <AD>
0888:E1 F5 F3 80 E6 E5 E3 EB <20>
0890:E5 F2 80 B0 B2 AF A7 B2 <E6>
0898:A1 AD AD A9 A5 B2 B4 80 <B8>
08A0:B7 AF B2 A4 A5 AE 8E 80 <E9>
08A8:80 80 80 80 88 E3 89 80 <23>
08B0:91 99 98 97 00 00 00 00 <DB>
08B8:00 00 00 00 24 32 35 25 <42>
08C0:23 2B 25 00 73 74 61 72 <72>
08C8:74 00 35 2D 00 21 2E 3A <26>
08D0:35 26 21 2E 27 25 2E 00 <F5>
08D8:00 00 00 00 00 00 00 A4 <8C>
08E0:B2 B5 A5 A3 AB A5 80 F3 <CF>
08E8:E5 EC E5 E3 F4 80 B5 AD <E9>
08F0:80 A2 A9 AC A4 A5 B2 80 <75>
08F8:BA B5 80 A5 B2 B3 B4 A5 <41>
0900:AC AC A5 AE 80 80 80 80 <45>
0908:80 80 80 80 80 80 80 80 <E5>
0910:80 80 80 80 80 80 80 80 <DD>
0918:80 80 80 B2 AF B4 AF B2 <C9>
0920:A5 AE 80 90 90 80 80 AC <5A>
0928:A5 B6 A5 AC 80 90 90 80 <6E>
0930:80 B0 B5 AE AB B4 A5 80 <A1>
0938:90 90 90 90 90 90 90 80 <87>
0940:80 80 80 B3 B0 B2 A9 B4 <A6>
0948:80 90 90 90 90 8A 8A 8A <84>
0950:8A 8A 8A 8A 8A 8A 8A 8A <8B>
0958:8A 8A 8A 8A 8A 8A 8A 8A <82>
0960:8A 8A 8A 8A 8A 8A 8A 8A <7A>
0968:8A A2 10 9D 42 03 A9 00 <E2>
0970:9D 48 03 A9 0B 9D 49 03 <BE>
0978:A9 00 9D 44 03 A9 64 9D <55>
0980:45 03 20 56 E4 60 A2 10 <53>
0988:A9 0C 9D 42 03 20 56 E4 <53>
0990:AD AE 04 8D 42 03 AD AF <5E>
0998:04 8D 43 03 AD B0 04 8D <29>
09A0:44 03 AD B1 04 8D 45 03 <16>
09A8:AD B2 04 8D 48 03 AD B3 <59>
09B0:04 8D 49 03 AD B4 04 8D <52>
09B8:4A 03 AD B5 04 8D 4B 03 <7F>
09C0:60 A9 FF 8D FC 02 20 3D <FA>
09C8:B2 A9 01 8D 8D 04 A9 A8 <34>
09D0:8D 9B 04 A9 02 8D 9C 04 <68>
09D8:A9 32 8D 80 04 A9 C9 8D <1A>
09E0:81 04 A9 E6 8D C0 02 A9 <BC>
09E8:AA 8D C1 02 A9 52 8D C5 <D8>
09F0:02 A9 0A 8D C6 02 A9 08 <06>
09F8:8D 97 04 A9 00 8D 92 04 <1F>
0A00:8D 86 04 8D 9A 04 8D 99 <56>
0A08:04 8D 93 04 20 0F B2 20 <B4>
0A10:42 B4 4C C3 B2 A0 00 B9 <F5>
0A18:C2 B7 99 20 06 C8 C0 32 <04>
0A20:D0 F5 AD 27 06 85 DA AD <F8>
0A28:28 06 85 DB AD 41 06 85 <13>
0A30:D6 AD 42 06 85 D7 AD 23 <E8>
0A38:06 8D B7 04 AD 2D 06 8D <84>
0A40:B8 04 60 A9 00 A0 00 99 <DF>
0A48:F6 53 99 F6 54 99 00 50 <0F>
0A50:99 00 51 99 00 52 99 00 <DE>
0A58:53 C8 C0 FF D0 E9 A0 00 <F0>
0A60:B9 52 B6 99 08 51 C8 C0 <9A>
0A68:C8 D0 F5 A0 00 B9 1A B7 <CD>
0A70:99 80 53 C8 C0 58 D0 F5 <54>
0A78:A0 00 B9 72 B7 99 28 50 <D7>
0A80:C8 C0 40 D0 F5 60 48 8A <41>
0A88:48 98 48 A9 50 AE F1 6E <07>
0A90:AC F0 6E 8D 0A D4 8D 09 <3F>
0A98:D4 8C 17 D0 8E 18 D0 AD <BB>
0AA0:F2 6E 8D 16 D0 AD F3 6E <A1>
0AA8:8D 19 D0 AD A5 04 8D 05 <71>
0AB0:D4 AE B7 04 AC B8 04 AD <2A>
0AB8:A8 04 8D 04 D4 8E 23 06 <B4>
0AC0:8C 2D 06 68 A8 68 AA 68 <C5>
0AC8:40 A9 00 8D 88 04 AD 78 <1F>
0AD0:02 8D 98 04 C9 0F D0 0C <8A>
0AD8:20 E3 B4 20 CD AD 20 75 <00>
0AE0:B3 4C C3 B2 20 75 B3 AE <DF>
0AE8:98 04 AD 81 04 AC 80 04 <5B>
0AF0:E0 0E F0 0F E0 0D F0 2C <27>
0AF8:E0 07 F0 49 E0 0B F0 52 <25>
0B00:4C C3 B2 C9 41 F0 55 18 <42>
0B08:AD 93 04 E9 27 8D 93 04 <3F>
0B10:18 AD 9B 04 E9 27 8D 9B <8A>
0B18:04 AD 9C 04 E9 00 8D 9C <0D>
0B20:04 4C 59 B3 C9 E1 F0 34 <47>
0B28:18 AD 93 04 69 28 8D 93 <DE>
0B30:04 18 AD 9B 04 69 28 8D <40>
0B38:9B 04 AD 9C 04 69 00 8D <4E>
```

»Heliman« (Anfang)

```
0B40:9C 04 4C 59 B3 C0 CE F0 <54>
0B48:13 EE 93 04 EE 9A 04 4C <43>
0B50:59 B3 C0 32 F0 06 CE 93 <62>
0B58:04 CE 9A 04 4C 59 B3 4C <E8>
0B60:9B B4 20 CD AD AD 98 04 <15>
0B68:C9 0E F0 1E C9 07 F0 23 <AB>
0B70:C9 0D F0 25 C9 0B F0 2A <7B>
0B78:4C 59 B3 AD FC 02 C9 FF <74>
0B80:D0 29 AD 84 02 C9 01 D0 <FC>
0B88:1F 60 CE 81 04 CE 81 04 <40>
0B90:4C 59 B3 EE 80 04 4C 59 <7E>
0B98:B3 EE 81 04 EE 81 04 4C <B9>
0BA0:59 B3 CE 80 04 4C 59 B3 <78>
0BA8:4C 6C B4 C9 08 F0 27 C9 <1A>
0BB0:0B F0 29 C9 1F F0 2B C9 <D2>
0BB8:1E F0 2D C9 1A F0 2F C9 <C3>
0BC0:18 F0 31 C9 00 F0 36 C9 <66>
0BC8:3E F0 2F C9 15 F0 47 C9 <2C>
0BD0:0C F0 40 4C 66 B4 20 1B <0E>
0BD8:B6 4C 66 B4 20 FA B5 4C <63>
0BE0:66 B4 EE F0 6E 4C 66 B4 <76>
0BE8:EE F1 6E 4C 66 B4 EE F2 <84>
0BF0:6E 4C 66 B4 EE F3 6E 4C <82>
0BF8:66 B4 4C 59 B4 20 FD B3 <58>
0C00:4C 66 B4 A9 04 20 03 A8 <E7>
0C08:A9 07 20 63 B1 20 80 B1 <A0>
0C10:4C 66 B4 4C 77 A9 20 16 <E7>
0C18:B4 4C 66 B4 18 A0 01 B1 <F7>
0C20:D8 69 01 C9 3A F0 05 91 <57>
0C28:D8 4C 42 B4 18 A9 30 91 <F7>
0C30:D8 A0 00 B1 D8 69 01 C9 <56>
0C38:3A F0 05 91 D8 4C 42 B4 <3B>
0C40:A9 30 91 D8 A0 01 91 D8 <4A>
0C48:18 A0 00 B1 D8 69 60 A0 <32>
0C50:40 91 DA 18 A0 01 B1 D8 <6D>
0C58:69 60 A0 41 91 DA 60 A9 <1A>
0C60:08 20 03 A8 A9 0B 20 63 <43>
0C68:B1 20 80 B1 A9 FF 8D FC <E2>
0C70:02 60 AD 81 04 C9 C8 90 <D2>
0C78:0E AC 93 04 B9 51 B8 8D <3A>
0C80:94 04 A0 23 91 D6 60 18 <E6>
0C88:AD 2D 06 6D 9B 04 85 D4 <E4>
0C90:AD 2E 06 6D 9C 04 85 D5 <25>
0C98:AD 94 04 AC 9A 04 91 D4 <3B>
0CA0:60 20 E3 B4 EE 88 04 AD <D3>
0CA8:88 04 C9 05 F0 03 4C 5C <8F>
0CB0:B3 4C C3 B2 CE 80 04 4C <71>
0CB8:E3 B4 EE 80 04 4C E3 B4 <F0>
0CC0:CE 81 04 CE 81 04 AD 88 <22>
0CC8:04 C9 01 D0 1C 20 FA B5 <3C>
0CD0:20 CD AD 4C E3 B4 EE 81 <C9>
0CD8:04 EE 81 04 AD 88 04 C9 <37>
0CE0:01 D0 06 20 1B B6 20 CD <DF>
0CE8:AD AE 81 04 E0 3F F0 E6 <74>
0CF0:E0 E3 F0 CC AD 80 04 C9 <24>
0CF8:31 F0 BF C9 CF F0 B5 8D <5D>
0D00:00 D0 8D 01 D0 A0 00 B9 <85>
0D08:3C B6 9D F6 53 E8 C8 C0 <6A>
0D10:0D D0 F4 E8 9D F6 53 AE <BB>
0D18:81 04 B9 3C B6 9D F6 54 <32>
0D20:E8 C8 C0 16 D0 F4 EE 87 <4E>
0D28:04 AD 87 04 C9 0E F0 19 <40>
0D30:EE 91 04 AD 91 04 C9 19 <38>
0D38:F0 52 EE 85 04 AD 85 04 <C8>
0D40:C9 80 F0 33 C9 FF F0 41 <34>
0D48:60 A9 00 8D 87 04 18 AD <66>
0D50:86 04 69 18 8D 86 04 C9 <2C>
0D58:60 F0 14 A2 00 AC 86 04 <68>
0D60:B9 BD AF 9D 08 53 C8 E8 <CA>
0D68:E0 18 D0 F4 4C 2A B5 A9 <4F>
0D70:00 8D 86 04 4C 55 B5 A0 <B8>
0D78:00 A2 00 BD B2 B7 99 08 <AC>
0D80:50 C8 C8 E8 C0 20 D0 F3 <7D>
0D88:60 4C DB B5 A9 00 8D 91 <8B>
0D90:04 18 AD 90 04 69 08 8D <A4>
0D98:90 04 C9 30 F0 50 AC 90 <C4>
0DA0:04 A9 00 A2 00 99 48 53 <4A>
0DA8:C8 E8 E0 08 D0 F7 EE 99 <0C>
0DB0:04 AD 99 04 C9 02 D0 0F <88>
0DB8:A9 00 8D 99 04 A2 08 A9 <C2>
0DC0:10 8D 97 04 4C C8 B5 A2 <C6>
0DC8:00 A9 08 8D 97 04 BD 1D <32>
0DD0:B0 99 48 53 C8 E8 EC 97 <B4>
0DD8:04 D0 F3 AD 9F B5 4C 34 <AC>
0DE0:B5 A9 00 A0 00 99 08 50 <7C>
0DE8:C8 C0 20 D0 F8 60 A9 00 <E7>
0DF0:8D 90 04 A0 00 99 78 53 <23>
0DF8:C8 C0 08 D0 F8 4C 98 B5 <39>
0E00:AD 2E 06 C9 6C D0 08 AD <2B>
0E08:2D 06 C9 48 D0 01 60 18 <35>
0E10:AD 2D 06 69 28 8D B8 04 <FD>
0E18:AD 2E 06 69 00 8D 2E 06 <05>
0E20:60 AD 2E 06 C9 64 D0 08 <3D>
0E28:AD 2D 06 C9 00 D0 01 60 <D4>
0E30:18 AD 2D 06 E9 27 8D B8 <3E>
0E38:04 AD 2E 06 E9 00 8D 2E <36>
0E40:06 60 00 00 00 00 00 30 <A7>
0E48:18 0C 06 03 01 00 00 00 <6C>
0E50:00 F8 F0 E0 C0 80 00 00 <66>
0E58:AA AA AA AA AA AA AA AA <B9>
0E60:AA AA 2A 0A 0A 0A 02 00 <3E>
0E68:AA AA A8 A8 A0 80 80 80 <53>
0E70:AA 2A 28 2A 0A 0A 2A 2A <0A>
0E78:80 80 A0 A0 A2 AA AA AA <9D>
0E80:02 0A 0A 2A 2A 2A AA AA <FB>
0E88:80 80 A0 A0 A8 A8 A8 AA <AC>
0E90:AA A8 A0 A0 A8 80 A0 AA <E6>
0E98:A8 A0 A0 A8 A0 AA A8 A8 <CE>
0EA0:00 80 82 A2 AA AA AA AA <DC>
0EA8:80 A0 A8 AA AA AA AA AA <3F>
0EB0:00 00 00 00 80 A0 A8 AA <44>
0EB8:AA AA AA 2A 28 08 08 08 <65>
0EC0:AA AA AA 28 20 00 00 00 <D4>
0EC8:2A 0A 02 02 15 FF C0 C0 <C1>
0ED0:A8 A0 80 80 54 FF 03 03 <98>
0ED8:00 20 20 28 28 2A AA AA <4A>
0EE0:2A 0A 0A AA 2A 0A 0A 2A <AF>
0EE8:A2 AA 8A AA AA AA A2 22 <39>
0EF0:00 00 00 00 02 0A 2A AA <42>
0EF8:02 0A 2A AA AA AA AA AA <6C>
0F00:00 8A 8A 8A 00 A8 A8 A8 <10>
0F08:00 00 00 00 08 28 AA AA <04>
0F10:AA AA A2 A0 A0 A0 80 80 <EA>
0F18:AA AA 8A 0A 0A 0A 02 02 <04>
0F20:FF AF AB AA AB AB AF AF <8D>
0F28:AF AF AB AA AA AA AA AA <52>
0F30:FF FF CE AA AA AA AA AA <DB>
0F38:FF FF BE AA AA AA AA AA <DD>
0F40:FF BF CF AA AA AA AA AA <FB>
0F48:FA EA EA AA AA AA AA AA <A6>
0F50:FF FA EA EA FA FA EA EA <BF>
0F58:FA EA EA FA EA AA EA EA <7E>
0F60:AB AB AF AB AF AB AA AB <71>
0F68:00 00 00 20 00 00 00 00 <88>
0F70:00 00 00 00 00 00 00 02 <90>
0F78:FF FF FF FF FF FF FF FF <96>
0F80:AB AB AB AB AB AB AB AB <33>
0F88:EA EA EA EA EA EA EA EA <A6>
0F90:FF AA AA AA AA AA AA AA <63>
0F98:AA AA AA AA AA AA AA FF <4C>
0FA0:00 0A 0A 0A 00 A8 A8 A8 <78>
0FA8:00 80 80 80 00 A8 A8 A8 <94>
0FB0:55 9A 9A 9A 55 A9 A9 A9 <06>
0FB8:01 04 10 40 40 10 04 01 <A1>
0FC0:04 04 04 04 00 11 00 00 <E6>
0FC8:70 70 46 EB B7 10 42 01 <91>
0FD0:B8 00 82 10 44 00 64 04 <0B>
0FD8:04 04 04 04 04 04 04 04 <F6>
0FE0:04 04 04 04 04 04 04 00 <7A>
0FE8:44 51 B8 04 04 04 41 20 <84>
0FF0:06 00 00 00 00 00 00 00 <11>
0FF8:65 64 69 74 6F 72 00 00 <83>
1000:00 00 00 00 00 00 00 80 <A0>
1008:80 80 80 80 80 80 80 80 <D7>
1010:80 80 80 80 80 80 80 80 <CF>
1018:80 80 80 80 80 80 80 80 <C7>
1020:80 80 80 80 80 80 80 80 <BF>
1028:80 80 80 80 80 80 80 80 <B7>
1030:80 80 80 80 80 80 80 80 <AF>
1038:80 80 80 80 80 A2 E9 EC <6C>
1040:E4 80 AE F2 80 9A 80 90 <38>
1048:90 80 80 80 80 80 80 80 <8F>
1050:80 80 80 80 80 80 80 A2 <B1>
1058:A1 A3 A6 B3 A7 B4 B5 A5 <AC>
1060:AA B7 B1 AB AC B8 AD AE <A9>
1068:B9 B2 A4 A8 A9 8A B6 8C <2D>
1070:8B F9 FA 80 80 80 80 80 <5C>
1078:F0 80 80 80 F6 80 80 AF <0A>
1080:B0 80 2F 30 00 00 00 00 <B1>
1088:00 00 00 00 00 00 00 00 <A8>
1090:00 00 00 00 00 00 00 00 <B0>
1098:00 00 00 00 00 00 00 00 <B8>
10A0:F8 80 80 80 F7 80 80 63 <A2>
10A8:00 E1 80 E2 80 85 80 80 <CC>
10B0:80 80 80 80 80 80 80 80 <2F>
10B8:80 80 80 80 80 80 80 80 <27>
10C0:80 80 80 80 80 80 80 80 <1F>
10C8:F1 F2 F3 F4 F5 80 80 01 <21>
10D0:00 02 00 03 00 04 04 00 <B9>
10D8:00 08 06 6A 6B 6C 6D 6E <A6>
10E0:6F 07 09 00 00 00 00 00 <EE>
```

»Heliman« (Schluß)

```
Programmname :BILD00.DAT
Länge        :2816 Bytes
---------------------------------------

0000:A1 A1 A1 A1 A1 A1 A1 A1 <C5>
0008:A1 A1 A1 A1 A1 A1 A1 A1 <BD>
0010:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0018:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0020:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0028:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0030:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0038:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0040:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0048:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0050:A1 A1 A1 A1 A1 A1 A1 A1 <76>
0058:A1 A1 A1 A1 A1 A1 A3 AD <7E>
0060:A2 A1 A8 AD AE AD AE A2 <3E>
0068:A1 A1 A1 A1 A1 B8 00 2F <46>
0070:30 B9 A1 A1 A1 A1 A1 A1 <00>
0078:A1 A1 A1 A8 B9 A1 A1 A1 <FC>
0080:A1 A1 A1 A1 A3 AD 00 00 <A3>
0088:00 A2 B8 00 00 00 00 00 <28>
0090:AE A2 A1 A1 A1 00 00 00 <83>
0098:00 00 AE B2 A1 A1 A1 A1 <DB>
00A0:A1 A1 A1 A1 A1 A1 A1 A1 <25>
00A8:A1 A1 A1 A1 A9 00 00 80 <B5>
00B0:80 80 80 80 80 80 00 00 <CE>
00B8:00 00 AE A1 A1 A7 00 00 <20>
00C0:00 00 00 A2 A1 A1 A1 A1 <5E>
00C8:A1 A1 A1 A1 A1 A1 A1 A1 <FD>
00D0:A1 A1 A1 A3 00 00 00 00 <A7>
00D8:00 00 80 00 00 00 00 00 <C8>
00E0:00 00 00 B2 B8 AD 00 00 <7C>
00E8:00 00 00 00 B2 A1 A1 A1 <E3>
00F0:A1 A1 A1 A1 A1 A1 A1 A1 <55>
00F8:A1 A1 B8 00 00 00 00 00 <A8>
0100:00 80 80 00 00 00 00 00 <32>
0108:00 00 00 A6 AB 00 00 00 <B9>
0110:00 00 00 00 A2 A1 A1 A1 <8B>
0118:A1 A1 A1 A1 A1 A1 A1 A1 <A3>
0120:A1 A3 00 00 00 00 00 00 <9B>
0128:00 80 80 00 00 00 00 00 <1A>
0130:00 00 00 B2 A8 00 00 00 <9A>
0138:00 00 00 00 80 B2 A1 A1 <E7>
0140:A1 A1 A1 A1 A1 A1 A1 A1 <8C>
0148:B8 00 00 00 00 00 00 00 <26>
0150:00 80 80 80 00 00 00 00 <6A>
0158:00 00 00 B2 A1 A7 B7 00 <82>
0160:00 00 00 00 00 A2 A1 A1 <CB>
0168:A1 A1 A1 A1 A1 A1 A1 A8 <9B>
0170:00 00 00 E2 04 04 04 04 <CC>
0178:04 04 04 E2 00 00 00 00 <1C>
0180:00 00 00 A2 A1 A1 A8 00 <85>
0188:00 00 00 00 00 00 B2 A1 <90>
0190:A1 A1 A1 A1 A1 A1 A1 AA <44>
0198:00 00 00 A1 00 00 00 00 <A4>
01A0:00 80 B5 A1 A5 00 00 00 <68>
01A8:00 00 00 00 AE A2 A1 A7 <8C>
01B0:B1 00 00 00 00 00 B2 A1 <0F>
01B8:A1 A1 A1 A1 A1 A1 A1 A1 <03>
01C0:A7 B4 B5 A8 00 00 00 00 <34>
01C8:00 00 B2 A1 A1 A7 00 00 <BE>
01D0:00 00 00 00 00 00 B2 A1 <D7>
01D8:A1 A7 00 00 00 00 A6 A1 <C1>
01E0:A1 A1 A1 A1 A1 A1 A1 A1 <EB>
01E8:A1 A1 A1 A3 00 00 00 00 <C1>
01F0:00 00 B9 A1 A1 A8 00 00 <BB>
01F8:00 00 00 00 00 00 AE A2 <F8>
0200:A1 A1 00 00 00 00 A2 A1 <21>
0208:A1 A1 A1 A1 A1 A1 A1 A1 <B1>
0210:A1 A1 A3 00 00 00 00 00 <61>
0218:00 00 00 B2 A1 A1 A7 00 <10>
0220:00 00 00 00 00 00 00 A6 <CA>
0228:A9 AD 00 00 00 00 B7 B2 <0B>
0230:A1 A1 A1 A1 A1 A1 A1 A1 <8A>
0238:A1 A1 AB AC 00 00 00 00 <05>
0240:00 00 00 B9 B9 A1 A8 00 <79>
0248:00 00 00 00 00 00 00 B9 <05>
0250:A8 80 00 00 00 00 B9 A1 <1C>
0258:A1 A1 A1 A1 A1 A1 A1 A1 <66>
0260:A1 A1 A1 A1 AB AC 00 00 <13>
0268:00 00 00 00 A6 A1 A1 A7 <06>
0270:00 00 00 00 00 00 00 00 <74>
0278:AD 80 00 00 00 00 00 A1 <93>
0280:A1 A1 A1 A1 A1 A1 A1 A1 <59>
0288:A1 A1 A1 B8 00 AD 00 00 <EB>
0290:00 00 00 00 B2 A1 A1 A1 <88>
0298:A7 E2 04 04 04 04 04 04 <E5>
02A0:04 04 04 04 04 04 E2 B2 <0E>
02A8:A1 A1 A1 A1 A1 A1 A1 A1 <51>
02B0:A1 A1 B8 00 00 00 00 00 <64>
02B8:00 00 00 00 A2 A1 A1 A1 <30>
02C0:A1 A8 00 00 00 00 00 00 <3F>
02C8:00 00 00 00 00 00 A2 A1 <B1>
02D0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
02D8:A1 A1 A5 B7 A7 00 00 00 <8A>
```

```
02E0:00 00 00 00 00 B9 A1 A1 <AE>
02E8:A1 A1 AB AC 00 00 00 00 <95>
02F0:00 00 00 00 00 00 B7 A1 <03>
02F8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0300:A1 A1 A1 A1 A7 00 00 00 <0D>
0308:00 00 00 00 00 00 B2 A1 <14>
0310:A1 A1 A1 A1 AB AC 00 00 <41>
0318:00 00 00 00 B4 A5 A1 A1 <32>
0320:A1 A1 A1 A1 A1 A1 A1 A1 <B7>
0328:A1 A1 A1 A1 A1 A5 00 00 <FC>
0330:00 00 00 00 00 00 E2 A4 <9F>
0338:A1 A1 A1 A1 A1 AB AC <9F>
0340:B4 B5 AB B5 A1 A1 A1 A1 <67>
0348:A1 A1 A1 A1 A1 A1 A1 A1 <70>
0350:A1 A1 A1 A1 A1 A7 00 <D3>
0358:00 00 00 00 00 01 00 A2 <04>
0360:A1 A1 A1 A1 A1 A1 A1 A1 <78>
0368:A1 A1 A1 A1 A1 A1 A1 A1 <90>
0370:A1 A1 A1 A1 A1 A1 A1 A1 <C7>
0378:A1 A1 A1 A1 A1 A1 B8 00 <4C>
0380:00 00 00 00 01 00 00 00 <8E>
0388:AD AE A2 A1 A1 A1 A1 A1 <8E>
0390:A1 A1 A1 A1 A1 A1 A1 A1 <27>
0398:A1 A1 A1 A1 A1 A1 A1 A1 <1F>
03A0:A1 A1 A1 A1 A1 A5 00 <9E>
03A8:00 00 00 01 00 00 00 00 <BE>
03B0:63 00 00 AD AE AE A2 A1 <3F>
03B8:A1 A1 A1 A1 A1 A1 A1 A1 <FF>
03C0:A1 A1 A1 A1 A1 A1 A1 A1 <17>
03C8:A1 A1 A1 A1 A1 A1 A1 A7 <F5>
03D0:00 00 01 00 00 00 00 00 <F6>
03D8:00 00 00 00 00 00 00 AD <8B>
03E0:A2 A1 A1 A1 A1 A1 A1 A1 <77>
03E8:A1 A1 A1 A1 A1 AD A1 A1 <3F>
03F0:A1 A1 A1 A1 A1 A1 A1 B8 <5E>
03F8:80 01 00 00 00 00 00 00 <FE>
0400:00 00 00 00 00 00 00 00 <08>
0408:00 AE AD A2 A1 A1 A1 A1 <E9>
0410:A1 A1 A1 A1 A1 A7 B9 A1 <F5>
0418:A1 A1 A1 A1 A1 A1 A1 B1 <B5>
0420:E2 00 00 00 00 00 00 00 <19>
0428:00 00 00 00 00 00 00 00 <30>
0430:00 00 00 00 A2 A1 A1 A1 <B6>
0438:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0440:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0448:A1 A7 AC 80 80 80 80 80 <50>
0450:80 80 80 80 80 80 80 80 <A7>
0458:80 80 80 80 80 AD A1 A1 <B5>
0460:A1 A1 A1 A1 A1 A1 A1 A1 <9E>
0468:A1 A1 A1 A1 A1 A1 A1 A1 <D5>
0470:A1 A1 A1 AB AC B4 AC 80 <05>
0478:80 B4 AC 80 80 80 80 80 <EC>
0480:80 80 80 80 B4 B5 A1 A1 <4B>
0488:A1 A1 A1 A1 A1 A1 A1 A1 <35>
0490:A1 A1 A1 A1 A1 A1 A1 A1 <2D>
0498:A1 A1 09 A1 A1 A1 AB <B2>
04A0:B5 A1 A1 AB AC 80 80 80 <2C>
04A8:80 80 80 80 E2 B9 A1 A1 <A5>
04B0:A1 A1 A1 A1 A1 A1 A1 A1 <0D>
04B8:A1 A1 A1 A1 A1 A1 A1 A1 <05>
04C0:A1 A1 A1 EA A1 A1 A1 A1 <81>
04C8:A1 A1 A1 A1 A1 AB AC 80 <12>
04D0:80 80 80 01 80 80 A2 A1 <A3>
04D8:A1 A1 A1 A1 A1 A1 A1 A1 <E5>
04E0:A1 A1 A1 A1 A1 A1 A1 A1 <1D>
04E8:A1 A1 A1 EB A1 A1 A1 A1 <F8>
04F0:A1 A1 A1 A1 A1 A1 A1 AB <57>
04F8:A7 80 01 80 80 80 B2 A1 <20>
0500:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0508:A1 A1 A1 A1 A1 A1 A1 A1 <BB>
0510:A1 A1 A1 EC A1 A1 A1 A3 <51>
0518:AD A2 A1 A1 B3 A1 A1 A1 <AE>
0520:AD 01 80 80 80 80 A2 A1 <C3>
0528:A1 A1 A1 A1 A1 A1 A1 A1 <9C>
0530:A1 A1 A1 B8 AD A2 A8 AD <74>
0538:B9 A1 A3 ED A2 A1 B8 80 <91>
0540:80 80 A2 A1 A1 A1 A1 B8 <13>
0548:01 80 80 80 80 80 B2 A1 <F1>
0550:A1 A1 A1 A1 B8 AE A2 A1 <51>
0558:A1 A1 A1 B8 80 80 AD 80 <47>
0560:80 AD 80 EE 00 AD 80 80 <5B>
0568:80 80 80 A2 A1 A1 A3 01 <FD>
0570:80 80 80 80 80 80 B2 A1 <0A>
0578:A1 A1 A1 A3 00 00 00 A2 <9B>
0580:A1 A8 AE 80 80 80 80 80 <94>
0588:80 80 80 EF 80 80 80 80 <56>
0590:80 80 80 80 A2 A3 01 80 <04>
0598:80 80 80 80 80 80 B9 A1 <EF>
05A0:A1 A1 A1 A7 00 00 80 80 <C3>
05A8:A2 B8 80 80 80 80 80 80 <2E>
05B0:80 80 80 EA 80 80 80 80 <EE>
05B8:80 80 80 80 00 01 80 80 <3F>
05C0:80 80 80 80 80 80 B2 A1 <B9>
05C8:A1 A1 A1 B8 00 80 80 80 <79>
05D0:80 80 80 80 80 80 80 80 <25>
05D8:80 80 80 EF 80 80 80 80 <06>
05E0:80 80 80 00 01 80 80 80 <21>
05E8:80 80 80 80 80 80 A6 A1 <7A>
05F0:B6 B6 B6 8B 00 80 80 80 <41>
05F8:80 80 80 80 80 80 80 80 <FD>
0600:80 80 80 EE 80 80 80 80 <DC>
0608:80 00 00 01 80 80 80 80 <E3>
0610:80 80 80 80 B1 B5 A1 A1 <A0>
0618:B6 B6 B6 8B 80 80 80 80 <DB>
0620:80 80 80 80 80 80 80 80 <D3>
0628:80 B1 80 ED 80 B1 80 80 <CF>
0630:00 00 01 80 80 80 80 80 <D3>
0638:80 80 B4 B5 A1 A1 A1 A1 <6F>
0640:B6 B6 B6 8B 80 80 80 80 <74>
0648:80 80 80 B7 80 80 B7 B4 <BB>
0650:B5 A1 A7 EC A6 A1 AB AC <2D>
0658:00 00 E2 80 80 B4 B5 A5 B4 <32>
0660:B1 AA A1 A1 A1 A1 A1 A1 <07>
0668:B6 B6 B6 8B 80 80 80 80 <EB>
0670:80 B4 B5 A1 AB B5 A1 A1 <FE>
0678:A1 A1 A1 EB A1 A1 A1 A1 <F4>
0680:A1 A1 A1 A1 A1 A1 A1 A1 <31>
0688:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0690:B6 B6 B6 8B 80 80 80 00 <E3>
0698:A6 A1 A1 A1 A1 A1 A1 A1 <D8>
06A0:A1 A1 A1 EA A1 A1 A1 A1 <EC>
06A8:A1 A1 A1 A1 A1 A1 A1 A1 <09>
06B0:A1 A1 A1 A1 A1 A1 A1 A1 <01>
06B8:B6 B6 B6 8B 80 80 80 8A <05>
06C0:B6 B6 A1 A1 A1 A1 A1 A1 <A1>
06C8:A1 A1 A1 08 A1 A1 A1 A1 <53>
06D0:A1 A1 A1 A1 A1 A1 A1 A1 <E1>
06D8:A1 A8 B9 A1 A1 A1 A1 A1 <B6>
06E0:B6 B6 B6 8B 80 80 80 8A <5D>
06E8:B6 B6 A1 A1 A1 A1 A1 A1 <19>
06F0:A1 A1 A1 A1 A1 A1 A1 A1 <41>
06F8:A1 A1 A1 A1 A1 A1 A1 A1 <D9>
0700:A1 A1 A1 A1 A1 A1 A1 A1 <AF>
0708:B6 B6 B6 8B 80 80 80 8A <D3>
0710:B6 B6 A1 A1 A1 A1 A1 A1 <90>
0718:A1 A1 A1 A1 A1 A1 A1 A1 <B7>
0720:A1 A1 A1 A1 A1 A1 A1 A1 <CF>
0728:A1 A1 A1 A1 A1 A1 A1 A1 <88>
0730:B6 B6 B6 8B 80 80 80 8A <8B>
0738:B6 B6 A1 A1 A1 A1 A1 A1 <28>
0740:A1 A1 A1 A1 A1 A1 A1 A1 <70>
0748:A1 A1 A1 A1 A1 A1 A1 A1 <68>
0750:A1 A1 A1 A1 A1 A1 A1 A1 <60>
0758:B6 B6 B6 8B 80 80 80 8A <E3>
0760:B6 B6 A1 A1 A1 A1 A1 A1 <80>
0768:A1 A1 A1 A1 A1 A1 A1 A1 <C7>
0770:A1 A1 A1 A1 A1 A1 A1 A1 <BF>
0778:A1 A1 A1 A1 A1 A1 A1 A1 <57>
0780:B6 B6 B6 8B 80 80 80 8A <38>
0788:B6 B6 A1 A1 A1 B3 A1 A1 <3F>
0790:A1 A1 A1 A1 A1 A1 A1 A1 <1F>
0798:A1 A1 A1 A1 A1 A1 A1 A1 <37>
07A0:A1 A1 A1 A1 A1 A1 A1 A1 <4F>
07A8:B6 B6 B6 8B 80 80 80 8A <33>
07B0:B6 B6 A1 A1 A1 A1 A1 A1 <AF>
07B8:A1 A1 A1 A1 A1 A1 A1 A1 <17>
07C0:A1 A1 A1 A1 A1 A3 A2 A1 <F9>
07C8:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
07D0:B6 B6 B6 8B 80 80 80 8A <2B>
07D8:B6 B6 A1 A1 A1 A1 A1 A1 <07>
07E0:A1 A1 A1 A1 A1 A1 A1 A1 <0F>
07E8:A1 A3 B2 A3 AD 80 80 A2 <A0>
07F0:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
07F8:B6 B6 B6 8B 80 80 80 8A <C3>
0800:B6 B6 A1 A1 A1 A1 A1 A1 <62>
0808:A1 A1 A1 B8 A1 A3 AE A2 <37>
0810:A3 80 AD 80 80 80 80 80 <E8>
0818:AE B9 AD A2 A1 A1 A1 A1 <D7>
0820:B6 B6 B6 8B 80 80 80 8A <A9>
0828:B6 B6 A1 A1 A1 A1 A1 A1 <3A>
0830:A1 A1 A3 80 AD 80 80 80 <29>
0838:80 80 80 80 80 80 80 80 <B7>
0840:80 80 80 80 B9 A1 A1 A1 <5C>
0848:B6 B6 B6 8B 80 80 80 8A <B1>
0850:B6 B6 A1 A1 A1 A1 A1 A1 <52>
0858:A1 AD 80 80 80 80 80 80 <7C>
0860:80 80 80 80 80 80 80 80 <8F>
0868:80 80 80 80 B2 A1 A1 A1 <FC>
0870:B6 B6 B6 8B 80 80 80 8A <49>
0878:B6 B6 A1 A1 A1 A1 A1 A1 <ED>
0880:A3 80 80 80 80 80 80 80 <DD>
0888:80 80 80 80 80 80 80 80 <67>
0890:80 80 80 80 A2 A1 A1 A1 <54>
0898:B6 B6 B6 8B 80 80 80 8A <21>
08A0:B6 B6 A1 A1 A1 A1 A3 AD <D5>
08A8:80 80 80 80 80 80 80 80 <47>
08B0:80 80 80 80 80 80 80 80 <3F>
08B8:80 80 80 80 80 B2 A1 A1 <61>
08C0:B6 B6 B6 8B 80 80 80 8A <09>
08C8:B6 B6 A1 A1 A1 A3 80 80 <82>
08D0:80 80 80 80 80 80 80 80 <1F>
08D8:80 80 80 80 80 80 80 80 <17>
08E0:80 80 80 80 80 A4 A1 A1 <01>
08E8:B6 B6 B6 8B 80 80 80 80 <47>
08F0:80 AD A2 A1 A3 80 80 80 <A4>
08F8:80 80 80 80 80 80 80 80 <F7>
0900:80 80 80 80 B1 80 80 80 <74>
0908:80 80 80 80 80 B9 A1 A1 <2B>
0910:B6 B6 B6 8B 00 00 00 00 <5A>
0918:00 00 00 AD 00 00 00 00 <F4>
0920:00 00 00 00 00 00 00 00 <32>
0928:00 00 B4 B5 A1 A5 A7 B7 <AB>
0930:00 00 00 00 00 00 8A B6 <0D>
0938:A1 F0 E1 E1 E1 E1 E1 E1 <F6>
0940:E1 E1 E1 E1 E1 E1 E1 E1 <96>
0948:E1 E1 E1 E1 E1 E1 E1 E1 <7E>
0950:E1 F6 A1 A1 A1 A1 A1 A8 <F1>
0958:00 00 00 00 00 00 8A B6 <35>
0960:A1 F8 E1 E1 E1 E1 E1 E1 <38>
0968:E1 E1 E1 E1 E1 E1 E1 E1 <9E>
0970:E1 E1 E1 E1 E1 E1 E1 E1 <03>
0978:E1 A1 A1 A1 A1 A1 A1 A1 <53>
0980:A7 00 00 00 00 00 8A B6 <31>
0988:A1 F8 E1 E1 E1 E1 E1 E1 <C2>
0990:E1 F6 F8 E1 E1 E1 E1 E1 <CC>
0998:E1 E1 F6 F0 E1 E1 E1 F6 <57>
09A0:F3 A1 A1 A1 A1 A1 A1 A1 <E6>
09A8:A8 00 00 00 00 00 8A B6 <59>
09B0:A1 F1 F3 F8 F6 F3 F1 F3 <7D>
09B8:F3 A1 A1 F4 F0 F1 F3 F0 <21>
09C0:F6 F3 F3 A1 F3 F4 F3 A1 <E1>
09C8:A1 A1 A1 A1 A1 A1 A1 A1 <E3>
09D0:A1 A7 00 00 00 00 8A B6 <A7>
09D8:A1 A1 A1 A1 A1 A1 A1 A1 <13>
09E0:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
09E8:A1 A1 A1 A1 A1 A1 A1 A1 <43>
09F0:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
09F8:A1 A1 A7 00 00 00 8A B6 <C1>
0A00:A1 A1 A1 A1 A1 A1 A1 A1 <A9>
0A08:A1 A1 A1 A1 A1 A1 A1 A1 <A1>
0A10:A1 A1 A1 A1 A1 A1 A1 A1 <B9>
0A18:A1 A1 A1 A1 A1 A1 A1 A1 <D1>
0A20:A1 A1 B6 8B 00 00 8A B6 <27>
0A28:A1 A1 A1 A1 A1 A1 A1 A1 <82>
0A30:A1 A1 A1 A1 A1 A1 A1 A1 <9A>
0A38:A1 A1 A1 A1 A1 A1 A1 A1 <72>
0A40:A1 A1 A1 B3 A1 A1 A1 A1 <89>
0A48:A1 A1 B6 8B 00 00 8A B6 <FF>
0A50:A1 A1 A1 A1 A1 A1 A1 B3 <8C>
0A58:A1 A1 A1 A1 A1 A1 A1 A1 <92>
0A60:A1 A1 A1 A1 A1 A1 A1 A1 <C9>
0A68:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0A70:A1 A1 B6 8B 00 00 8A B6 <F6>
0A78:A1 A1 A1 A1 A1 A1 A1 A1 <31>
0A80:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0A88:A1 A1 A1 A1 A1 A1 A1 A1 <21>
0A90:A1 A1 A1 A1 A1 A1 A1 A1 <39>
0A98:A1 A1 B6 8B 00 00 8A B6 <EE>
0AA0:A1 A1 A1 A1 A1 A1 A1 A1 <09>
0AA8:A1 A1 A1 A1 A1 A1 A1 A1 <01>
0AB0:A1 A1 A1 A1 A1 A1 A1 A1 <19>
0AB8:A1 A1 A1 A1 A1 A1 A1 A1 <F1>
0AC0:A1 A1 B6 8B 00 00 8A B6 <87>
0AC8:A1 A1 A1 A1 A1 A1 A1 A1 <E1>
0AD0:A1 A1 A1 A1 A1 A1 A1 A1 <F9>
0AD8:A1 A1 A1 A1 A1 A1 A1 A1 <11>
0AE0:A1 A1 A1 A1 A1 A1 A1 A1 <49>
0AE8:A1 A1 B6 8B 00 00 8A B6 <DA>
0AF0:B2 DF 46 F2 00 00 00 00 <FC>
0AF8:00 00 00 00 00 00 00 00 <0C>
0B00:00 00 00 00 00 00 00 00 <16>
```

»Level 1« zum Abtippen

```
Programmname :BILD01.DAT
Länge        :2816 Bytes
------------------------------------

0000:A1 A1 A1 A1 A1 A1 A1 A1 <C5>
0008:A1 A1 A1 A1 A1 A1 A1 A1 <BD>
0010:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0018:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0020:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0028:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0030:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0038:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0040:A1 A1 A1 A1 A1 B8 A1 A1 <B3>
0048:A1 B9 A1 A1 A1 A1 A1 A1 <70>
0050:A1 A1 A1 A1 A1 A1 A1 A1 <76>
0058:A1 A1 A1 A1 A1 A1 A1 A1 <6E>
0060:A1 A1 A1 A1 A1 A1 A1 A1 <62>
0068:A1 A1 A1 A1 A3 00 2F <F1>
0070:30 00 A2 A1 A1 A1 A1 A1 <D0>
0078:A1 A1 A1 A1 A1 A1 A1 A1 <CD>
0080:A1 A1 A1 A1 A1 A1 A1 A1 <45>
```

»Level 2« (Anfang)

Ergänzen Sie jetzt Ihre HAPPY-COMPUTER-Sammlung

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle »Happy Computer«-Ausgaben von 1985? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?
Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der noch lieferbaren Ausgaben. Und so kommen Sie schnell an die gewünschten Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

Stichwort	Titel	Seite/Ausgabe
	Aktuelles	
Computer	Amiga — ein Traumcomputer wird Wirklichkeit	9/10
	Atari: Lage gefestigt	14/11
	Der »Plus/4« ist endlich da	12/2
	Grundstein einer neuen Linie und kein zweiter PC	13/10
	Konsequentes Chaos (Der deutsche QL)	14/10
DFÜ	Akustikkoppler für C 64	9/1
	Ascom-Koppler jetzt auch für Atari	20/8
	Ein Anschluß unter dieser Nummer (Mailbox Nummern)	159/3
	Mailboxbetrieb in den USA	22/10
	Neues DFÜ-Programm für den Spectrum	22/10
	Nullmodem zum Aufstecken	12/1
Software	Atari-Schreiber jetzt für 520 ST	14/12
	Software fast zum Nulltarif	10/1
	Träume werden wahr (Schneider-Neuheiten aus England)	9/12
Drucker	Mac Inker, der sparsame Drucker	12/12
Floppy	Commodore-Floppy auf Trab gebracht	9/1
	Diskettenlaufwerk für den Sharp MZ-800	12/1
	Quick Disk — Die Floppy-Alternative (MSX)	20/4
Erweiterung	Mini-Expansion-Box für TI 99/4A	11/1
MSX	Das Musikwunder (Yamaha CX-5)	141/2
	Der Billig-MSX von Philips kommt	60/1
	CP/M mit MSX-Computer: so geht's	141/2
	Ein komplettes System von Philips	19/8
	Flotter Dreier (Sanyo, Goldstar und Canon)	23/5
	MSX-Mix	45/3
	Mit dem fliegenden Teppich auf Erfolgskurs	15/10
Bücher	Bücher zur DFÜ	111/3
	Bücher zum Denken (KI)	120/10
	Messeberichte	
	Die neuesten Heimcomputer (Winter-CES)	9/3
	Funkausstellung in Berlin: MSX war Trumpf	9/11
	Kampf der Kolosse (Winter-CES — Teil 1)	9/4
	Sommer-CES 1985: Weiche Welle in Chicago — Teil 1	9/8
	Software-Jackpot (Winter CES — Teil 2)	9/5
	Software-Super-Show in London (PCW-Show)	12/11
KI	Künstliche Intelligenz in Wiesbaden (AI Europa)	13/12
Musik	Musikmesse Frankfurt: Midi marschiert	22/5
	Interviews	
	David Crane (Ghostbusters Autor)	17/5
	Interview mit den »Print Shop«-Machern	14/8
	Jack Tramiel (Chairman Atari)	11/2
	Hardware-Tests	
Drucker	Bewußt robust (Europrint K 6311 FT)	31/5
	Software-Tests	
Textverarb.	Ein Textprogramm, das sich lohnt (Homeword/C 64)	77/4
	Drei Drucker im Test (STX 80, Gemini 10X, CP-80X)	16/1
	(Nachhall auf Seite 149 in 4/85) DWX 305: Schönschrift zum Niedrigpreis	18/2
	Eine heiße Verbindung (EP 22, EP 44, EXD 10)	26/5
	Kompakt und leise: Matrixdrucker GLP (Centronics)	24/1
	Regenbogenfarben — wie gedruckt (Okimate 20)	154/10
	Schön oder schnell (Horizon HX 80)	21/3
	Spectrum mit starken Typen (Gabriele 9009)	126/11
	Zwei Drucker für den Schneider (NLQ 401, GP 500 CPC)	112/8
Computer	Chinese mit britischem Paß (Triton 64)	22/2
	Der Musik Maestro (Yamaha CX-5)	28/4
	Der Neue: Commodore PC 128	46/5
	Der »neue« Spectrum	31/1
	Ein »Einsteiger« aus Taiwan (BIT-90)	16/2
	Joyce — Schneiders Einstieg in die Welt der PCs	24/11
	Quantensprung im Schneckentempo (QL dt. Version)	180/11
	Koreaner mit Deutsch-Talent (Ce-Tec/MSX)	18/3
	Schneiders neue Dimension (CPC 6128)	24/10
	Sharps Jüngster (Sharp MZ-800)	20/1
	Spectrum plus oder Spectrum minus	24/4
	Viel Computer für wenig Geld (Schneider CPC 664)	113/8
	YC-64: Fernöstlicher Biedermann (MSX Computer)	20/2
	Wer ist wer? (Atari 520 ST+ und 260 ST)	16/12
	Wie musikalisch ist mein Heimcomputer?	148/11
Laufwerke	3-Zoll-Erfahrungen (MCD-1-Floppy für Spectrum) (Discovery/Spectrum)	22/1
	Ein ungleiches Paar (Spectrum — VIC 1541 Interface)	21/4
	Lauf, Floppy, lauf! (SpeedDos plus/C64)	45/12
	Preiswertes Spectr um Floppysystem (Viscount System)	21/2
	Spectrum Diskettensystem im Plus-Look	20/3
	VC 1541 wird zur Rennfloppy	42/4
Recorder	Der Spectrum Sprinter (Datenrecorder: Sprint)	28/1
	Ein billiger Speicher für alle (Recorder MC 3810)	30/5
DFÜ	DFÜ auch mit dem TI (RS 232 für TI 99/4A)	25/5
	Kommunikation mit dem Spectrum	32/4
	Spartanisch aber gut (Ascom Akustikkoppler)	168/3
Sonstiges	Computer steuert Modelleisenbahn	176/11
	Der andere Weg (Spectrum Tastatur)	19/3
	Faszination der Technik (Fischer Technik Roboter)	44/11
	Famose Formel für den C64 (Formel 64)	40/12
	Grafpad Supergrafik für den Spectrum	16/3
	Haltet den Dieb (Alarmanlage für C 64, VC 20)	29/1
	Ohren oder Tasten? (Voice Command Modul/C 64)	40/10
	Peripherie für MSX (Plotter, 3½-Zoll-Floppy) (Joysticks im Vergleichstest)	26/1
	Roboter, Technologie der Zukunft (Fischertechnik)	45/4
	Starker Arm für Heimcomputer (Teach Robot)	38/4
	Tafelfreuden für Grafik-Gourmets (Atari Maltafel)	14/1/
	Vom Piepmatz zum Mini-Orchester (Spectrum Sound)	15/2
	Software-Tests	
Textverarb.	Ein Textprogramm, das sich lohnt (Homeword/C 64)	77/4
	Jedem seine Zeitung (The Newsroom)	118/8
	Schreiben ohne Frust	46/2
	Textverarbeitung für jedermann (Homewriter für MSX)	137/1
Sprachen	Basic-Erweiterung zum Spartarif (Aztec Basic/C 64)	76/4
	Drei Assembler für Atari-Computer im Vergleich	30/3
	Fortschritt rückwärts (CP/M-80 Emulator für 520 ST)	138/11
	Hisoft-Pascal jetzt Microdrive-kompatibel (Spectrum)	56/2
	Logo für den Atari 520 ST	134/11
	Mallard-80-Basic — ein starkes Stück	28/11
	Maschinensprache ist keine Zauberei (CPC 464)	107/8
	Prozessor-Welt von morgen: C 64 simuliert 68000	42/10
	Spezielles Spiele-Basic für den Spectrum	143/5
	Welches Basic für meinen MZ-700?	48/2
	Zwölf Farben in Mode 2 (Color Star für CPC 464)	110/8
Utilities	Das Programm, das Programme macht (Progressor)	33/5
	Disketten-Doktor für den C 128	42/12
	Quicksave für Spectrum	137/4
	SM-Kit — Das Werkzeug für Lehrling und Meister (C 64)	138/1
	Software-Knackern dazwischengepfuscht (Apple II)	27/3
Grafik	Beeindruckend (Print Shop — Druckprogramm)	50/2
	Die Maus bringt Farbe auf den Bildschirm (Apple)	52/2
	Grafik grandios (Malprogramm Blazing Paddles)	28/3
	Koala Bilder zum Anfassen (Hardcopy-Programm)	57/2
	Mit dem Joystick programmiert (Designers Pencil)	140/5
	Viel Grafik für wenig Geld (Graphics Basic und Supergrafik 64 für C 64 im Vergleich)	44/2
	Vorsicht Kamera! (Take 1, Trickfilm Designer)	126/8
DFÜ	Apple II sucht Anschluß	154/3
	Contact 64 — Die Software zum Ascom-Koppler	142/5
	Spectrum auf Draht (DFÜ Vergleichstest)	124/8
Astronomie	Spectrums Sternstunden	34/3
	Sterngucker	158/10
Schach	Schachmatt per Telefon	156/10
	Spiele-Tests	
	Amazon	145/5
	Archon II: Adept	126/2
	Asylum	144/3
	Athletic Land	146/1
	A View to a Kill	169/10
	Ballblazer	167/10
	Boulder Dash	125/2
	Bounty Bob strikes back	139/8
	Cavelord	124/2
	Crazy Train	144/1
	D-Bug	118/2
	Deus ex Machina	146/4
	Don't buy this	168/12
	Doomdark's Revenche	148/5
	Dorodon	142/3
	Dragonsden	124/2
	Elektro Freddy	145/1
	Elite	164/10
	Eureka	144/4
	Fahrenheit 451	145/5
	Five-a-Side Football	166/10
	Formula One	140/8
	Frank Brunos Boxing	166/10
	Frankie goes to Hollywood	162/10
	Fruity Frank	145/4
	Ghettoblaster	169/11
	Ghostbusters	138/3
	Ghost Chaser	170/11
	Great American Cross Country Road Race	168/11
	Hacker	167/12
	H.E.R.O.	149/5
	Hyper Sports 1	143/3
	Karateka	146/4
	Kennedy Approach	168/12
	Knight Lore	143/3
	Macbeth	144/4
	Mask of the Sun	122/2
	Match Day	150/5
	Mindshadow	141/8
	Mr. Do	167/10
	Monster Trivia	168/10
	Nick Faldo plays the Open	169/11
	Nightshade	169/12
	Nodes of Yesod	169/12
	On Court Tennis	150/5
	Pitfall II	148/5
	Rama	145/5
	Rescue on Fractalus	168/10
	Rocket Ball	140/8
	Rockford's Riot (Bolder Dash II)	168/11
	Rock'n Bolt	139/8
	Sherlock Homes	121/2
	Seastalker	147/1
	Serpent's Star	142/4
	Software Star	165/11
	Spelunker	142/3
	Standing Stones	145/4
	Summer Games II	133/8
	Super Pipeline II	141/8
	The Ancient Art of War	149/5
	The Fourth Protocol	165/11
	The Hitchhiker's Guide to the Galaxy	138/4
	The Little Computer People Projekt	170/12
	The Way of exploding Fist	169/10
	Tour de France	170/11
	Where in the World is Carmen San Diego	163/11
	Whistler's Brother	141/3
	White Lightning	148/1
	Winter Games	164/12
	World Championship Boxing	170/12
	ZimSalaBim	141/3
	Spiele Tips	
	Abenteuer im Weltraum	152/5
	Amazon	172/10
	Aztec Challenge	147/4
	Aztec Tomb	147/4
	Aztec Tomb	173/10
	Beach Head	85/1
	Death in the Caribbean	142/8
	Death in the Caribbean	172/12
	Doomdark's Revenge	142/8
	Everyone's a Wally	173/10
	Ghostbusters	140/3
	Ghostbusters	147/4
	Ghostbusters	152/5
	Hampstead	172/12
	Heros of Karn	173/12
	Hexenküche	173/12
	Hobbit	146/3
	Hulk	143/8
	Hunch Back	85/1
	Karateka	172/12
	Lode Runner	174/11
	Lode Runner	174/12
	Mask of the Sun	173/11
	Masquerade	144/8
	Mindshadow	174/11
	Miner 2049er	147/4
	Pirate Adventure	126/2
	Pitfall	147/4
	Pitfall II	144/8
	Pitfall II	174/10
	Sabre Wulf	85/1
	Sands of Egypt	173/11
	Sands of Egypt	174/12
	Schloß des Grauens	152/5
	Secret Mission	173/11
	Ship of Doom	172/12
	Spelunker	144/8
	Strip Poker	152/5
	Summer Games	144/8
	Summer Games	172/10
	Super Huey	171/10
	The Dallas Quest	174/10
	The Institute	172/10
	The Institute	173/12
	The Quest	173/11
	The Witness	146/3
	Time Maschine	173/11
	Ultima II	126/2
	Ultima II	151/5
	Ultima III	151/5
	Valhalla	173/11
	Whistler's Brother	144/8
	Zeppelin	152/5
	ZimSalaBim	142/8
	Zork	172/10
	Listings	
Anwendung	Alle Neune (Jahresauswertung-Kegeln/C 64)	67/5
	(Adreßverwaltung/C 64) Besseres Basic ganz einfach Software Basic 3.0/C 64)	67/3
	Datenbank mit freiem Zugriff (C 64)	86/1
	Der Halleysche Komet kommt (MSX)	76/10
	Die Mini-Textverarbeitung (Spectrum)	74/8
	Nachhall auf Seite 160 in 9/85	
	Do-it-yourself-Datenverwaltung (Mainfile II/C 64)	63/3
	Einblick ins Innenleben (Disassembler/CPC 464)	86/5
	Eine tolle Textverarbeitung für den Schneider (464)	90/3
	Nachhall auf Seite 85 in 5/85	
	Funktionen optisch aufbereitet (VZ-200/Laser)	95/3
	Geregelte Finanzen mit dem Commodore 64	80/11
	Nachhall auf Seite 117 in 8/85	87/4
	Morse-Decoder für Funkamateure (Spectrum)	117/11
	Nebenkostenabrechnung (C 64)	63/10
	Nachhall auf Seite 80 in 12/85 Optik mit Simons Basic (C 64)	63/3
	Programme in Reih' und Glied (C 64)	62/12
	Suchen, nein danke (Dateiverwaltung/CPC 464)	83/4
	Nachhall auf Seite 117 in 8/85	
	Transistor-Schaltungen berechnen (L.d.M./Spectrum)	51/1
	Turbo-Basic-Interpreter für Atari 800XL (L.d.M.)	81/12
Grafik	Apple IIc-HiRes-Grafik auf dem Drucker	105/11
	Bewegte Grafik mit drei Befehlen (CPC 464)	74/10
	Farbspielereien (Atari)	89/3
	Grafikentzerrung für Matrixdrucker (Spectrum)	108/12
	Grafik-Window bekommt Nachwuchs (C 64)	68/10
	Nachhall auf Seite 80 in 12/85	
	Grafikzauber (Apple II)	78/2
	Hires Fantasy (C 64)	58/3
	Rosetten-Grafik für den Spectrum	88/3
	Schnelle Grafik aus dem Compiler (L.d.M./C 64)	49/8
	Schöne schnelle Grafik (Grafik-Paket/C 64)	80/2
	Solar-Painter (Spectrum)	101/12
	Sprite-Editor (C 64)	82/1
	Zauber der Farben mit Magic Painter (L.d.M./Atari)	83/3
	Nachhall auf Seite 85 in 5/85	
	Zaubereien auf dem Bildschirm (L.d.M./Grafik/CPC 464)	80/4
	Zeichenroutine für Kreise und Ellipse (CPC 464)	90/5
	Zykloide für Grafiker und Mathematiker (C 64)	60/10
	Nachhall auf Seite 79 in 12/85	
Spiel	Das Haus des Magiers (C 64)	63/4
	Dasher, der Volltreffer (L.d.M./C 64)	62/5
	Nachhall auf Seite 117 in 8/85	
	Der rasende Raider (C 64)	79/3
	Diamantenfieber (L.d.M./Atari 48 KByte)	58/2
	Nachhall auf Seite 85 in 5/85	
	Die Abenteuer eines rasenden Reporters (Report/C 64)	60/1
	Geröllheimer (Atari)	79/5
	Gespensterjagd im Schneider (CPC 464)	74/2
	Nachhall auf Seite 85 in 5/85	
	Lumberjack Larrys Abenteuer in Bagdad (S.d.M/C 64)	52/12
	Kalte Zeiten (Wintry Screen/C 64)	69/2
	Kneipe zum hastigen Kellner (VC 20)	68/1
	Mit dem Apple auf die Trainerbank (Aktion Apfelsaft)	100/4
	Mit dem Atari-Computer auf Ölsuche (Atari)	68/8
	Mücke mit Tücke (C 64)	70/1
	Nachtflug (Spectrum)	72/1
	Nachhall auf Seite 85 in 5/85 Niemandsland (C 64)	72/3
	Pokerface für 16 KByte (Spectrum)	76/5
	Psycho — die Macht des Geistes (C 64)	64/8
	Nachhall auf Seite 80 in 12/85	
	Rennfahrer mit dem Joystick (Driver/C 64)	71/4
	Rettet den letzten Baum (Insekt defense/C 64)	72/2
	SAM — der Mann von der Baustelle (L.d.M./CPC 464)	109/11
	Nachhall auf Seite 79 in 12/85	
	Schatzhöhle (Atari 800XL)	75/1
	Vorsicht Hochwasser (Aquantor/L.d.M./C 64)	54/10
	Über den Wolken (Flugplanung/C 64)	73/11
	Wortsuchspiel (Spectrum)	104/3
Tips&Tricks	AMPEL — grünes Licht für Atari-Maschinen-Programme	104/12
	Auf Trap gebracht (CPC 464)	73/12
	Basic bequem (C 64)	85/2
	Basic-Compactor (Spectrum)	82/10
	Nachhall auf Seite 80 in 12/85	
	Basic-Plus: Applesoft-Basic-Erweiterung (Apple II)	96/3
	Beim C 64 piepst es	70/11
	Bilder richtig konservieren (CPC 464)	97/4
	Bildschirmtrick für den Commodore 64	70/11
	Byte-Shifter (Spectrum)	91/5
	Dateien hin- und hergerissen (Atari-IBM)	94/4
	Der neue Checksummer ist da (C 64)	64/10
	Der neue Checksummer ist da (C 64)	69/11
	Der neue Checksummer (C 64)	67/12
	Deutsche Sonderzeichen unter CP/M (CPC 464)	69/12
	Die Maltafel wird zur Maus (Atari)	98/4
	Nachhall auf Seite 49 in 11/85	
	Disk-Help für die schnelle Hilfe (Atari)	71/8
	Disk- und DOS-Utility für alle Atari-Computer	77/10
	Drei Tricks für MSX	92/5
	Ein langes Gesicht für den C 64 (Longscreen 64)	72/11
	Fehlerhilfe mit HELP & TRACE (VC 20)	95/1
	Fensterlkünstler (C 64)	71/5/
	Fettschrift für den 48 KByte-Spectrum	98/5
	Find Label (Spectrum)	88/2
	Flotte Primzahlen in Hisoft Pascal (Spectrum)	86/2
	Fußball-Manager für Commodore 64	58/8
	Grafik-Hardcopy in vierfacher Größe (C 64)	61/8
	Kostenlose Speichererweiterung (C 64)	67/10
	Listen leicht gemacht (C 64)	94/1
	Make DATA für den Spectrum	102/3
	Maschinencode-Routinen in Basic umgesetzt (CPC 464)	75/10
	Microdrive-Aufwertung (Spectrum)	122/11
	Nachhall auf Seite 79 in 12/85	
	Mondlandung (C 64)	56/8
	Musik und Farbe (C 64)	68/12
	Nie mehr Listingkummer mit dem Checksummer (C 64)	84/2
	Nie mehr Listingkummer mit dem Checksummer (C 64)	61/4
	Prima Werkzeug für den Programmierer (C 64)	59/8
	Programmtransfer leicht gemacht (CPC 464)	72/10
	Proportionalschrift für den Spectrum	94/5
	Protokoll auf dem Drucker (CPC 464)	76/8
	RAM-Disk für Atari 800XL	119/11
	Ran an den Userport (C 64)	72/11
	Renumber 64 (C 64)	70/5
	Nachhall auf Seite 117 in 8/85	
	Rock me Amadeus (C 64)	66/11
	RSX-Befehle ohne »@« (Schneider)	73/12
	Schilderwald (Plakatschrift/C 64)	65/10
	Schluß mit der Eintönigkeit (C 64)	70/11
	Nachhall auf Seite 80 in 12/85	
	Spectrums COPY besser nutzen	83/10
	Spectrumtasten mit Funktionen belegt (Spectrum)	98/1
	Spectrum Tips & Tricks	106/12
	Sprachkurs für Commodore-Basic (C 64)	90/1
	Super-Merge für Commodore 64	54/8
	Nachhall auf Seite 160 in 9/85	
	Super-Saver (C 64)	67/10
	Statuszeile mit Uhr (Atari)	96/1
	Tasword 464 mit DIN-Tastatur (CPC 464)	73/10
	Texte auch im Grafikmodus (Atari)	92/5
	Tippen mit dem Plotter (C 64)	94/1
	Tips & Tricks rund um den Schneider	77/8
	Töne aus dem Atari	96/5
	Variablendump für Atari (Atari)	83/2
	Variablen-Transfer (Spectrum)	123/11
	Verflixter Listschutz (C 64)	68/10
	Nachhall auf Seite 80 in 12/85	
	Vom Maschinencode zum Basic-Programm (C 64)	66/10
	Wie die Bilder laufen lernten (Atari)	86/2
	Zeilenakrobatik auf dem Schneider	78/8
	Zwei SCREEN$ im schnellen Wechsel (Spectrum)	73/8
	ZX81-Utility: Nützliches für Aufsteiger (C 64)	61/11
	30 tolle Maschinencode-Routinen (Spectrum)	98/3
	Grundlagen	
Speicher	Daten am laufenden Band	26/8
	Daten auf der schnellen Scheibe	30/8
	Floppy gegen Kassette	38/8
	Selbst geschraubt ist halb gespart	39/8
	So arbeitet das 1050-Laufwerk von Atari	36/8
	So liest und schreibt die 1541	34/8
	Speichermedium Endlosband	28/8
	Tips, Tricks und Todsünden	41/8
	Wohin in Zukunft mit Bits und Bytes	24/8
Monitore	Farbmonitore — buntes Fenster zum Computer	127/5
	Monitore: Richtig geplant, gekauft und genossen	133/5
Drucker	Blitzsaubere Schrift mit Laserlicht (Laserdrucker)	147/10
	Die »heißen« Drucker (Thermodrucker)	139/10
	Farbspiele für Farbdrucker	135/10
	Mit leisen Tönen (Tintenstrahldrucker)	128/10
	Scharfe Nadel, spitze Typen (Matrix und Typenrad)	145/10

Stichwort	Titel	Seite/Ausgabe
prachen	Auf einen Blick: Logo-Befehle	132/2
	Befehlserweiterung für RSX (CPC 464)	34/10
	CP/M — Ein Betriebssystem	84/8
	Fenster in die Zukunft: Basic auf dem 520 ST	132/12
	Logo-Spielerei oder ernsthafte Alternative	110/1
	RSX — Maschinensprache mit Komfort	34/11
FÜ	Begriffe aus der DFÜ	161/3
	Datenübertragung im schnellen Gleichschritt	144/11
usik	Beethoven — Bit für Bit	152/11
	Der Weg zum Kabelorchester	157/11
onstiges	Das Interface 1 ROM und seine Nutzung	158/4
	Der Commodore 64 kann einfach alles	59/4
	Der 18er und sein RAM	43/12
	Ein großes Abenteuer: Das Adventure	128/2
	Messen + Steuern = Regeln	146/11
	Schnittstellen — was sind das eigentlich	36/4
	So bauen die Spiele-Baukästen	32/5
	Vom Traum zum Heimcomputer (68000 Prozessor)	20/11
	Weiche Hardcopy (Schneider)	74/12
	Welcher Computer spielt am besten?	158/12
	1, 2, 3 — Kalkulieren mit der Hand ist nun vorbei	80/8
	Allgemeine Themen	
ernen	Der Computer — Ein moderner Trichter?	116/2
	Schule mit Computer	118/10
FÜ	Keine Angst vor DFÜ	153/3
onstige	Amiga Spiele Premiere	161/12
	Bits auf Abwegen	147/11
	Computer als Briefträger	148/3
	Der C 64 im C 128	51/11
	Ein teures Vergnügen (DFÜ-Kosten)	154/3
	Happy-Sportspielführer	137/8
	Heimcomputer aus zweiter Hand	142/12
	Mehr als ein Computer (Die Commodore Story)	49/4
	Raupkopierer gegen den Rest der Welt	126/10
	Software (fast) geschenkt	151/10
	Software-Piraterie	23/8
	Software-Volltreffer	144/12
	Software zum Spartarif	153/10
	Spiele auf der schwarzen Liste	160/11
	Vom Heimcomputer-Freak zum EDV-Spezialisten	35/2
	Vom Hobby zum Geldregen	39/2
	Vom Abenteuer, ein Abenteuer zu schreiben	42/2
	Wenn mal was schiefgeht	140/12
	Wissenswertes, Fragen und Antworten zum 128er	52/11
	Zubehör und Software — das »kleine« Geschenk	39/1/
	Zu viel Kontrolle	150/3/
	1985 — Das Jahr der Eisenbahn	164/4
	Kurse	
ogo	Teil 1: Der Einstieg für Einsteiger	40/3
	Teil 2: Die Schildkröte lernt laufen	151/4
	Teil 3: Die Schildkröte wird erwachsen	153/5
ascal	Pascal für Schüler und Lehrer	86/8
	Pascal für kluge Köpfe/Teil 2	121/10
	Pascal für kluge Köpfe/Teil 3	124/11
tari	Schnelle Grafik für Atari Computer	124/10
64	Musik mit Poke und Peek/Teil 1	54/3
	Musik mit Poke und Peek/Teil 2	53/4
	Musik mit Poke und Peek/Teil 3	56/5
	Lernen Sie Ihren Commodore 64 kennen/Teil 1	59/5
	Lernen Sie Ihren Commodore 64 kennen/Teil 4	45/8
	Lernen Sie Ihren Commodore 64 kennen/Teil 6	45/10
	Lernen Sie Ihren Commodore 64 kennen/Teil 7	56/11
	Ohne Fleiß kein Kreis/Teil 1	48/12
PC 464	Kein Buch mit sieben Siegeln/Teil 1	156/5
	Kein Buch mit sieben Siegeln/Teil 4	105/8
ardware	Zugüberwachung per Computer/Teil 1	155/4
	Zugüberwachung per Computer/Teil 2	51/5
	Basteln	
	Atari 520 ST auf Abwegen	23/12
	Bilder aus dem Weltall (Schneider)	32/12
	Dem User Port geht ein Licht auf (C 64)	54/11
	Fehler in der Spectrum Hardware	43/8
	Gute Verbindung mit dem Schneider (PIO-Interface)	28/10
	Lightshow mit dem Commodore 64	44/5
	Multitalent für den Joystickanschluß (Spectrum)	30/2
	Nachhall auf Seite 85 in 5/85	
	Nachhall auf Seite 77 in 7/85	
	Neue Geräteadresse für das 1541 Laufwerk (C 64)	62/10
	Nie wieder Angst (Alarmanlage C 64)	48/3
	Nachhall auf Seite 80 in 12/85	
	Schalten und walten mit dem Atari (Schaltinterface)	114/10
	Schreiben mit Schreibmaschinenqualität (C 64)	26/2
	Schreibschutz-Schalter (Atari 810 Floppy)	24/3
	Schreibschutz-Schalter (Atari 1050 Floppy)	107/11
	Sieben auf einen Port (7 Segment Anzeige/Spectrum)	24/2
	Sparen am richtigen »Drucker-Ende« (Sinclair)	23/3
	Verbesserte Cursorsteuerung beim Spectrum	29/2
	Zwei Joysticks für ein Halleluja (CPC 464)	31/5
	Marktübersichten	
99/4A	Erweiterungen zum TI 99/4A	40/1
ari	Marktübersicht Atari	46/1
	Rund um den Atari	128/11
ari ST	Jede Menge Software	132/11
ectrum	Anschluß gesucht: Peripherie für ZX81 und Spectrum	48/1
64	Interfaces für den Commodore 64	49/1
	Der Computer mit dem großen Zubehör	56/4
kustikkoppler	Akustikkoppler, preiswert wie noch nie	160/3
rucker	Druckerparade	129/10
	Nachhall auf Seite 80 in 12/85	
onitore	Marktübersicht Monitore	136/5
	Nachhall auf Seite 80 in 12/85	
usik	Musiksoftware	151/11
ftware	Softladen (Die neusten Programme und ihre Preise)	32/1
	So viel Software (Heimsoftware für Heimcomputer)	150/12
iele	Spiele aus dem Baukasten (Construction Sets)	38/5
omputer	Welcher Computer zum Weihnachtsfest?	136/12
	Wettbewerbe	
ifruf	Aktion Apfelsaft	29/1
ıflösung	Bildergalerie	106/1
ıflösung	Bildergalerie (Nachlese)	142/2
ifruf	Bithoven-Festival	46/3
ıflösung	Bithoven-Festival	128/8
ifruf	Der Computer als Steuermann	48/11
ifruf	Der schönste Titel von 1984	106/1
ıflösung	Der schönste Titel von 1984	135/5
ıflösung	Diskettenwettbewerb	176/10
ifruf	Happy Computer Leserwettbewerb	130/8
ıflösung	Happy Computer Leserwettbewerb	20/12
ifruf	Ihr Einsatz (Die beste Anwendung)	104/1
ifruf	Leserumfrage — Taschenrechner	70/10
ifruf	Probleme auf der Wörsalm	179/11
ıflösung	Spiel des Jahres	148/4
ifruf	Steno mit dem Computer	41/5
ifruf	Was steuern, wie regeln?	46/11
ıflösung	Wer gewinnt den goldenen Besenstiel	172/11
	Leserforum	
	Atari-Tips	102/1
	Autostart für VC 20	103/1
	Basicode-2 für MZ-700	77/2
	Basic-Speicher ohne Boden (C 64)	185/11
	Basic und HiRes-Grafik (C 64)	160/12
	Commodore-Ecke	117/10
	Eingabezeile beim Spectrum speichern	110/3
	Gedächtnislücke beim ZX 81	35/4
	gtext 64 an RX 80 angepaßt	103/1
	Joystickprobleme beim VC 20	77/2
	LPRINT III — Fehlerloses Drucken auch ohne EPROM	159/12
	Probleme mit den langen Zeilen (C 64)	185/11
	Probleme mit 800XL	159/12
	Sprite-Kollision (C 64)	160/12
	Stereo aus dem Commodore 64	110/3
	Tip für Oric 1	103/1
	Unvollständige Adresse beim ZX 81	77/2
	VC 20 und Videokamera am Monitor	103/1

Die Ausgaben 6/85, 7/85 und 9/85 sind bereits vergriffen und nicht mehr lieferbar!

Auch die bisher erschienenen Sonderhefte können Sie jetzt direkt bestellen:

SONDERHEFT 01/84: SINCLAIR
Unentbehrliche Informationen zu den Sinclair Computern ZX81 und Spectrum.

SONDERHEFT 01/85: SPECTRUM
Anwendungsbezogene Listings und Tips&Tricks für alle Spectrum-Fans.

SONDERHEFT 02/85: SCHNEIDER 1
Eine Fülle wertvoller Beiträge und Listings für alle Schneider-Anwender.

SONDERHEFT 03/85: SPIELE
Ein Super-Nachschlagewerk für alle Spiele-Fans mit 100 Spielen im Test und großer Marktübersicht.

SONDERHEFT 01/86: SCHNEIDER 2
Noch mehr Tips und Tricks für Einsteiger und Fortgeschrittene mit vielen interessanten Programm-Listings.

SONDERHEFT 02/86: ATARI 1
Besonders 800 XL- und 130 XE-Fans erwarten jede Menge Anwendungs- und Spiele-Listings sowie Informationen.

SONDERHEFT 03/86: 68000er
Umfassende Informationen zur neuen Computer-Generation und eine große Vergleichstabelle, die im Detail über alle 68000er informiert.

SONDERHEFT 04/86: SCHNEIDER 3
Eine Erweiterung für alle Schneider-Anwender, Super-Programm-Listings und großer Einsteiger-Teil.

SONDERHEFT 05/86: PROGRAMMIERSPRACHEN
Fuß fassen in »Pascal«, »C« und »Forth« mit jeweils einem grundlegendem Kurs und vielen Anwendungs-Listings.

SONDERHEFT 06/86: 68000er 2
Umfangreicher Listingteil, viele Informationen, Tips und Tricks für Anwender der 68000er-Computer.

SONDERHEFT 07/86: SCHNEIDER 4
Mit den Schwerpunkten Joyce und CP/M plus, Ratschlägen zur Vortex-Karte und vielen Tips & Tricks.

SONDERHEFT 08: COMPUTER ALS HOBBY
Wissenswertes für Einsteiger und zusätzliche Informationen zur Fernsehsendung Computerzeit.

SONDERHEFT 09: 68000er 3
Mit den Schwerpunkten Sound- und Videodigitalisierung und Spieleprogrammierung.

SONDERHEFT 10/86: SCHNEIDER 5
Der neue Schneider-PC wird vorgestellt. Wieder viele Hilfestellungen und Kurse.

SONDERHEFT 11/86: SPIELE-TESTS
Alles über aktuelle Spieletests, Computerprogramme, Grafik- und Musik-Software.

SONDERHEFT 12/86: 68000er 4
Ausführliche Testreihe aller Grafikprogramme für Atari ST, Amiga und Sinclair QL sowie viele Grundlageninformationen zu diesen Computern.

SONDERHEFT 13: SCHNEIDER 6
Diskettengrundlagen-Kurs. Reiche Auswahl an Programmen für CPC. Schneider CPC oder PC: Fakten und Vorteile.

SONDERHEFT 14: SOFTWARE
Der Softwareführer 1987 für Ihre optimale Programmauswahl.

SONDERHEFT 15: HARDWARE-TESTHEFT
Über hundert Geräte für optimale Hardware-Auswahl im Test.

SONDERHEFT 16: SCHNEIDER 7
Das Super-Programm CPC-Giga-Cad: Dreidimensionales Zeichnen plus animierte Grafik.

SONDERHEFT 17: SPIELE-TESTS
Ausgewählte Spieleneuerscheinungen vorgestellt und kritisch unter die Lupe genommen.

SONDERHEFT 18: SCHNEIDER 8
So schlägt Basic alles: kürzer, schneller, besser programmieren mit Plan.

SONDERHEFT 19: ATARI ST
Alles für Umsteiger, Programmierer und Assembler-Freaks.

Tragen Sie die Nummer des gewünschten Sonderheftes (z.B. 08/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein.

Am besten gleich mitbestellen: Die Happy-Computer-Sammelboxen



Für alle Leser, die »Happy Computer« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es ein interessantes Service-Angebot: die Happy-Computer-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk. Ein kompletter Jahrgang (12 Ausgaben) paßt in eine der praktischen Sammelboxen!

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

```
0088:A1 A1 A1 A1 A1 A1 A1 A1 <3D>
0090:A1 A1 A1 A8 AE 00 00 00 <AD>
0098:00 00 B2 A1 A1 A1 A1 A1 <4C>
00A0:A1 A1 A1 A1 A1 A1 A1 A1 <25>
00A8:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
00B0:A1 A1 A1 A1 A1 A1 A1 A1 <15>
00B8:A1 A1 A1 A3 00 00 00 00 <BF>
00C0:00 00 A6 A1 A1 A1 A1 A1 <1D>
00C8:A1 A1 A1 A1 A1 A1 A1 A1 <FD>
00D0:A1 A1 A1 A1 A1 A1 A1 A1 <F5>
00D8:A1 A1 A1 A1 A1 A1 A1 A1 <ED>
00E0:A1 A1 A3 00 00 00 00 00 <AD>
00E8:00 00 A4 A1 A1 A1 A1 A1 <F5>
00F0:A1 A1 A1 A1 A1 A1 A1 A1 <55>
00F8:A1 A1 A1 A1 A1 A1 A1 A1 <4D>
0100:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
0108:A3 AD 00 00 00 00 00 00 <07>
0110:00 00 A4 A1 A1 A1 A1 A1 <04>
0118:A1 A1 A1 A1 A1 A1 A1 A1 <A3>
0120:A1 A1 A1 A1 A1 A1 A1 A1 <AB>
0128:A1 A1 A1 A1 A1 A3 AD AE <00>
0130:00 00 00 00 00 00 00 00 <32>
0138:00 A6 A1 A1 A1 A1 A1 A1 <53>
0140:A1 A1 A1 A1 A1 A1 A1 A1 <8C>
0148:A1 A1 A1 A1 A1 A1 A1 A1 <74>
0150:A1 A1 A3 AE AD 00 00 00 <87>
0158:00 00 00 00 00 00 00 B7 <11>
0160:A6 A1 A1 A1 A1 A1 A1 A1 <F9>
0168:A1 A1 A1 A1 A1 A1 A1 A1 <94>
0170:A1 A1 A1 A1 A1 A1 A1 A1 <DB>
0178:AE AD 00 00 00 00 00 00 <BC>
0180:00 00 00 00 00 B4 B5 A1 <5B>
0188:A1 A1 A1 A1 A1 A1 A1 A1 <33>
0190:A1 A1 A1 A1 A1 A1 A1 A1 <3B>
0198:A1 A1 B8 A3 E2 A2 A1 A3 <30>
01A0:00 00 00 00 00 00 00 00 <A2>
01A8:00 00 B7 B4 B5 A1 A1 A1 <D7>
01B0:A1 A1 A1 A1 A1 A1 A1 A1 <1B>
01B8:A1 A1 A1 A1 A1 A1 A1 A1 <03>
01C0:B8 A3 00 00 03 00 AE 00 <BA>
01C8:00 00 00 00 00 00 00 00 <CA>
01D0:B4 B5 A1 A1 A1 A1 A1 A1 <65>
01D8:A1 A1 A1 A1 A1 A1 A1 A1 <E3>
01E0:A1 A1 A1 A1 A1 A1 A1 B8 <02>
01E8:00 00 00 00 03 00 00 00 <03>
01F0:00 00 00 00 00 B4 B5 A7 <D2>
01F8:B9 A1 A1 A1 A1 A1 A1 A1 <57>
0200:A1 A1 A1 A1 A1 A1 A1 A1 <D9>
0208:A1 A1 A1 A1 A1 A1 A3 00 <14>
0210:00 00 00 00 03 00 00 00 <2C>
0218:00 00 00 00 A6 A1 A1 A1 <B2>
0220:A1 A1 A1 A1 A1 A1 A1 A1 <B9>
0228:A1 A1 A1 A1 A1 A1 A1 A1 <D1>
0230:A1 A1 A1 A1 A1 B8 00 00 <03>
0238:00 00 00 00 03 00 00 00 <54>
0240:00 00 B4 B5 A1 A1 A1 A1 <8F>
0248:A1 A1 A1 A1 A1 A1 A1 A1 <72>
0250:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0258:A1 A1 A1 A1 A1 A9 00 00 <A2>
0260:00 00 00 00 E2 00 00 B4 <27>
0268:B5 A5 A1 A1 A1 A1 A1 A1 <C8>
0270:A1 A1 A1 09 A1 A1 A1 A1 <43>
0278:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0280:A1 A1 A1 A1 A1 A3 00 00 <7F>
0288:00 00 B4 B5 A1 AB B5 A1 <1A>
0290:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0298:A1 A1 A1 6F A1 A1 A1 09 <6C>
02A0:A1 A1 A1 A1 A1 A1 A1 A1 <39>
02A8:A1 A1 A1 A8 AD 00 00 00 <B9>
02B0:00 00 A2 A1 A1 A1 A1 A1 <66>
02B8:B8 B9 A3 AD AD AE AD AE <2B>
02C0:A2 A1 B8 6E B9 A1 A1 6A <D1>
02C8:A1 A1 A1 A1 A1 A1 A1 A1 <F1>
02D0:A1 A1 A8 A2 00 00 00 00 <6C>
02D8:00 00 00 A2 A1 A1 A1 A9 <6A>
02E0:00 00 00 00 00 00 00 00 <E4>
02E8:00 AD 00 6D 00 A2 A1 6B <21>
02F0:A2 A1 B9 A1 A1 A1 A1 A1 <CD>
02F8:A1 A1 A1 B1 00 00 00 00 <D4>
0300:00 00 00 00 A2 A1 A1 B8 <8E>
0308:00 00 00 00 00 00 00 00 <0E>
0310:00 00 00 6C 00 00 AD 6C <A2>
0318:00 AE 00 63 A2 A1 A1 A1 <35>
0320:A1 A1 A1 A1 A7 00 00 00 <82>
0328:00 00 00 00 00 A2 A3 00 <FB>
0330:00 00 00 00 00 00 00 00 <36>
0338:00 00 00 6B 00 00 00 6D <51>
0340:00 00 00 00 00 A2 A1 A1 <AF>
0348:A1 A1 A1 A1 A1 A1 A1 F0 <BF>
0350:E1 E1 E1 E1 E1 E1 E1 E1 <9A>
0358:E1 E1 E1 E1 E1 F6 A7 00 <5F>
0360:00 00 00 00 6A 00 6E 00 <8C>
0368:00 00 00 00 00 00 B9 A1 <82>
0370:A1 A1 A1 A1 A1 A1 A1 F8 <1E>
0378:E1 E1 E1 E1 E1 E1 E1 E1 <82>
0380:E1 E1 E1 E1 E1 F7 A1 A7 <A6>
0388:00 00 00 00 00 6F 00 00 <48>
0390:00 00 00 00 00 00 B2 A1 <9C>
0398:A1 A1 A1 A1 A1 A1 A1 F8 <76>
03A0:E1 F7 F0 E1 E1 E1 E1 E1 <16>
03A8:E1 E1 E1 E1 E1 F7 A1 A1 <48>
03B0:F0 E1 E1 E1 E1 E1 E1 E1 <42>
03B8:E1 E1 F6 00 00 00 B9 A1 <3A>
03C0:A1 A1 A1 A1 A1 A1 A1 F1 <67>
03C8:A1 A1 A1 F2 F5 F2 F0 F2 <DD>
03D0:F2 F5 F2 F5 F2 A1 A1 A1 <90>
03D8:F8 E1 E1 E1 E1 E1 E1 E1 <6E>
03E0:E1 E1 F7 00 00 00 B2 A1 <74>
03E8:A1 A1 A1 A1 A1 A1 A1 A1 <0F>
03F0:A1 A1 A1 A1 A1 A1 A1 A1 <47>
03F8:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
0400:F8 E1 E1 E1 E1 E1 E1 E1 <D3>
0408:E1 E1 F7 00 00 00 A4 A1 <C0>
0410:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0418:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0420:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0428:F1 A1 F4 F0 F4 F5 F4 F0 <99>
0430:F4 F5 A1 00 00 00 A2 A1 <67>
0438:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0440:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0448:A1 A1 A1 A1 A1 A1 A1 A1 <76>
0450:A1 A1 A1 A1 A1 A1 A1 A1 <6E>
0458:A1 A1 A3 80 80 80 B2 A1 <27>
0460:A1 A1 A1 A1 A1 A1 A1 A1 <9E>
0468:A1 A1 A1 A1 A1 A1 A1 A1 <D5>
0470:A1 A1 A1 A1 A1 A1 A1 A1 <CD>
0478:A1 A1 A1 A1 A1 A1 A1 A1 <45>
0480:A1 A8 80 80 80 80 B9 A1 <6F>
0488:A1 A1 A1 A1 A1 A1 A1 A1 <35>
0490:A1 A1 A1 A1 A1 A1 A1 A1 <2D>
0498:A1 A1 A1 A1 A1 A1 A1 A1 <25>
04A0:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
04A8:A1 A3 80 80 80 80 B2 A1 <FC>
04B0:A1 A1 A1 A1 A1 A1 A1 A1 <0D>
04B8:A1 A1 A1 A1 A1 A1 A1 A1 <05>
04C0:A1 A1 A1 A1 A1 A1 A1 A1 <FD>
04C8:A1 A1 A1 A1 A1 A1 A1 B8 <0C>
04D0:A3 80 80 80 80 80 A4 A1 <FE>
04D8:A1 A1 A1 A1 A1 A1 A1 A1 <E5>
04E0:A1 A1 A1 A1 A1 A1 A1 A1 <1D>
04E8:A1 A1 A1 A1 A1 A1 A1 A1 <55>
04F0:A1 A1 A1 A1 A1 A1 A3 80 <30>
04F8:80 80 80 80 80 80 B2 A1 <84>
0500:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0508:A1 A1 A1 A1 A1 A1 A1 A1 <BB>
0510:A1 A1 A1 A1 A1 A1 A1 A1 <A3>
0518:A1 A1 A1 A1 A1 A3 80 80 <51>
0520:80 80 80 80 00 A6 A1 A1 <CB>
0528:A1 A1 A1 A1 A1 A1 A1 A1 <9C>
0530:A1 A1 A1 A1 A1 A1 A1 A1 <84>
0538:A1 A1 A1 A1 A1 A1 A1 A1 <8C>
0540:A1 A1 A1 B8 A3 80 80 80 <0C>
0548:80 80 80 B7 A6 A1 A1 A1 <2F>
0550:A1 A1 A1 A1 A1 A1 A1 A1 <64>
0558:A1 A1 A1 A1 A1 A1 A1 A1 <6C>
0560:A1 A1 A1 A1 A1 A1 A1 A1 <94>
0568:A1 A1 A8 80 00 80 80 80 <AB>
0570:80 80 A6 A1 A1 A1 A1 A1 <4C>
0578:A1 A1 A1 A1 A1 A1 A1 A1 <4B>
0580:A1 A1 A1 A1 A1 A1 A1 A1 <33>
0588:A1 A1 A1 A1 A1 A1 A1 A1 <3B>
0590:A1 B8 A3 80 80 80 80 80 <22>
0598:80 B7 B9 A1 A1 A1 A1 A1 <23>
05A0:A1 A1 A1 A1 A1 A1 A1 A1 <53>
05A8:A1 A1 A1 A1 A1 A1 A1 A1 <1B>
05B0:A1 A1 A1 A1 A1 A1 A1 A1 <03>
05B8:A3 80 80 80 80 80 80 80 <AB>
05C0:A6 A1 A1 A1 A1 A1 A1 A1 <71>
05C8:A1 A1 A1 A1 A1 A1 A1 A1 <FB>
05D0:A1 A1 A1 A1 A1 A1 A1 A1 <E3>
05D8:A1 A1 A1 A1 A1 A1 A1 A3 <ED>
05E0:80 80 80 80 80 80 80 B7 <4C>
05E8:B2 A1 A1 A1 A1 A1 A1 A1 <C4>
05F0:A1 A1 A1 A1 A1 A1 A1 A1 <43>
05F8:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
0600:A1 A1 A1 A1 A1 A3 AD 80 <B0>
0608:80 80 80 80 80 B7 A6 A1 <33>
0610:A1 A1 A1 A1 A1 A1 A1 A1 <A1>
0618:A1 A1 A1 A1 A1 A1 A1 A1 <B9>
0620:A1 A1 A1 A1 A1 A1 A1 A1 <D1>
0628:A1 A1 A1 A1 A8 80 80 80 <DB>
0630:80 80 80 80 A6 A1 A1 A1 <DA>
0638:A1 A1 A1 A1 A1 A1 A1 A1 <9A>
0640:A1 A1 A1 A1 A1 A1 A1 A1 <72>
0648:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0650:A1 A1 A1 A1 A8 80 80 80 <B3>
0658:80 80 80 A6 A1 A1 A1 A1 <EA>
0660:A1 A1 A1 A1 A1 A1 A1 A1 <92>
0668:A1 A1 A1 A1 A1 A1 A1 A1 <C9>
0670:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0678:A1 A1 A1 A9 AD 80 80 80 <52>
0680:80 80 A6 A1 A1 A1 A1 A1 <29>
0688:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0690:A1 A1 A1 A1 A1 A1 A1 A1 <21>
0698:A1 A1 A1 A1 B8 A1 B9 A1 <20>
06A0:A1 A1 A1 A1 B1 80 80 80 <EA>
06A8:80 80 B2 A1 A1 A1 A1 A1 <80>
06B0:A1 A1 A1 A1 A1 A1 A1 A1 <01>
06B8:A1 A1 A1 A1 A1 A1 A1 A1 <19>
06C0:A1 A1 A1 A3 80 63 80 A2 <D2>
06C8:A1 A1 A1 A1 A8 80 80 80 <3B>
06D0:80 80 B2 A1 A1 A1 A1 A1 <78>
06D8:A1 A1 A1 A1 A1 A1 A1 A1 <F9>
06E0:A1 A1 A1 A1 A1 A1 A1 A1 <11>
06E8:A1 A1 A3 80 80 80 80 80 <8D>
06F0:A2 A1 A1 A1 A8 80 80 80 <12>
06F8:80 80 B9 A1 A1 A1 A1 A1 <0F>
0700:A1 A1 A1 A1 A1 A1 A1 A1 <AF>
0708:A1 A1 A1 A1 A1 A1 A1 A1 <A7>
0710:A1 A8 80 80 80 80 80 80 <47>
0718:80 AD A2 A1 A3 80 80 80 <77>
0720:80 80 B2 A1 A1 A1 A1 A1 <0A>
0728:A1 A1 A1 A1 A1 A1 A1 A1 <88>
0730:A1 A1 A1 A1 A1 A1 A1 A1 <80>
0738:A1 AD 80 80 80 80 80 80 <5E>
0740:80 80 80 AE 80 80 80 80 <9E>
0748:80 80 B2 A1 A1 A1 A1 A1 <E0>
0750:A1 A1 A1 A1 A1 A1 A1 A1 <60>
0758:A1 A1 A1 A1 A1 A1 A1 A1 <78>
0760:A1 A7 80 80 80 80 80 80 <F7>
0768:80 80 80 80 80 80 80 80 <89>
0770:80 80 B9 A1 A1 A1 A1 A1 <BD>
0778:A1 A1 A1 A1 A1 A1 A1 A1 <57>
0780:A1 A1 A1 A1 A1 A1 A1 A1 <2F>
0788:A1 A1 A7 80 80 80 80 80 <EB>
0790:80 80 80 80 80 80 80 80 <61>
0798:80 80 AD B8 A1 A1 A1 A1 <65>
07A0:A1 A1 A1 A1 A1 A1 A1 A1 <4F>
07A8:A1 A1 A1 A1 A1 A1 A1 A1 <07>
07B0:A1 A1 A1 A5 A7 80 80 80 <88>
07B8:80 80 80 80 80 80 80 80 <39>
07C0:80 80 80 A6 A1 A1 A1 A1 <8C>
07C8:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
07D0:A1 A1 A1 A1 A1 A1 A1 A1 <DF>
07D8:A1 A1 A1 A1 A1 AB AC B4 <48>
07E0:AC 80 80 80 80 80 80 80 <7B>
07E8:80 80 A6 A1 A1 A1 A1 A1 <BF>
07F0:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
07F8:A1 A1 A1 A1 A1 A1 A1 A1 <D7>
0800:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0808:A1 AB B5 A7 80 80 80 80 <13>
0810:80 80 B2 A1 A1 A1 A1 A1 <14>
0818:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0820:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0828:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0830:A1 A1 A1 A1 A7 80 80 80 <CF>
0838:80 80 A2 A1 A1 A1 A1 A1 <F2>
0840:A1 A1 A1 A1 A1 A1 A1 A1 <76>
0848:A1 A1 A1 A1 A1 A1 A1 A1 <6E>
0850:A1 A1 A1 A1 A1 A1 A1 A1 <62>
0858:A1 A1 A1 A1 A3 80 80 80 <C7>
0860:80 80 B2 A1 A1 A1 A1 A1 <C6>
0868:A1 A1 A1 A1 A1 A1 A1 A1 <CD>
0870:A1 A1 A1 A1 A1 A1 A1 A1 <45>
0878:A1 A1 A1 A1 A1 A1 A1 A1 <3D>
0880:A1 A1 A1 A1 A9 80 80 80 <8E>
0888:80 80 AD A1 A1 A1 A1 A1 <0C>
0890:A1 A1 A1 A1 A1 A1 A1 A1 <25>
0898:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
08A0:A1 A1 A1 A1 A1 A1 A1 A1 <15>
08A8:A1 A1 A1 A1 A8 80 80 80 <5F>
08B0:80 80 B1 B2 A1 A1 A1 A1 <63>
08B8:A1 A1 A1 A1 A1 A1 A1 A1 <FD>
08C0:A1 A1 A1 A1 A1 A1 A1 A1 <F5>
08C8:A1 A1 A1 A1 A1 A1 A1 A1 <ED>
08D0:A1 A1 A1 A1 B8 80 80 80 <B6>
08D8:80 80 A2 A1 A1 A1 A1 A1 <52>
08E0:A1 A1 A1 A1 A1 A1 A1 A1 <55>
08E8:A1 A1 A1 A1 A1 A1 A1 A1 <4D>
08F0:A1 A1 A1 A1 A1 A1 A1 A1 <C5>
08F8:A1 06 6A 6B 6C 6D 6E 6F <99>
0900:6A 6B 6C 6D 6E 07 A1 A1 <E0>
0908:A1 A1 A1 A1 A1 A1 A1 A1 <A3>
0910:A1 A1 A1 A1 A1 A1 A1 A1 <AB>
0918:A1 A1 A1 A1 A1 A1 A1 A1 <D3>
0920:A1 A1 A1 A1 B8 00 00 00 <EE>
0928:00 00 B2 A1 A1 A1 A1 A1 <FE>
0930:A1 A1 A1 A1 A1 A1 A1 A1 <8C>
0938:A1 A1 A1 A1 A1 A1 A1 A1 <74>
0940:A1 A1 A1 A1 A1 A1 A1 A1 <7C>
0948:A1 A1 A1 A1 A9 00 00 00 <3E>
0950:00 00 B9 A1 A1 A1 A1 A1 <11>
0958:A1 A1 A1 A1 A1 A1 A1 A1 <94>
0960:A1 A1 A1 A1 A1 A1 A1 A1 <DB>
0968:A1 A1 A1 A1 A1 A1 A1 A1 <C3>
0970:A1 A1 A1 A1 A8 00 00 00 <1F>
0978:00 00 B2 A1 A1 A1 A1 A1 <4E>
0980:A1 A1 A1 A1 A1 A1 A1 A1 <3B>
0988:A1 A1 A1 A1 A1 A1 A1 A1 <23>
0990:A1 A1 A1 A1 A1 A1 A1 A1 <2B>
0998:A1 A1 A1 A1 A3 00 00 00 <FE>
```

```
09A0:00 00 B2 A1 A1 A1 A1 A1 <66>
09A8:A1 A1 A1 A1 A1 A1 A1 A1 <03>
09B0:A1 A1 A1 A1 A1 A1 A1 A1 <0B>
09B8:A1 A1 A1 A1 A1 A1 A1 A1 <F3>
09C0:A1 A1 A8 A2 B8 00 00 00 <28>
09C8:00 00 B2 A1 A1 A1 A1 A1 <9A>
09D0:A1 A1 A1 A1 A1 A1 A1 A1 <EB>
09D8:A1 A1 A1 A1 A1 A1 A1 A1 <13>
09E0:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
09E8:A1 A1 A1 B1 00 00 00 00 <D2>
09F0:00 00 A2 A1 A1 A1 A1 A1 <D6>
09F8:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0A00:A1 A1 A1 A1 A1 A1 A1 A1 <A9>
0A08:A1 A1 A1 A1 A1 A1 A1 A1 <A1>
0A10:A1 A1 A1 A1 A7 80 80 80 <03>
0A18:00 00 B1 A1 A1 A1 A1 A1 <CA>
0A20:A1 A1 A1 A1 A1 A1 A1 A1 <8A>
0A28:A1 A1 A1 A1 A1 A1 A1 A1 <82>
0A30:A1 A1 A1 A1 A1 A1 A1 A1 <9A>
0A38:A1 A1 A1 A1 E2 04 04 04 <2E>
0A40:04 04 E2 A1 A1 A1 A1 A1 <17>
0A48:A1 A1 A1 A1 A1 A1 A1 A1 <66>
0A50:A1 A1 A1 A1 A1 A1 A1 A1 <7A>
0A58:A1 A1 A1 A1 A1 A1 A1 A1 <92>
0A60:A1 A1 A1 A1 E2 04 04 04 <86>
0A68:04 04 E2 A1 A1 A1 A1 A1 <2F>
0A70:A1 A1 A1 A1 A1 A1 A1 A1 <59>
0A78:A1 A1 A1 A1 A1 A1 A1 A1 <31>
0A80:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0A88:A1 A1 A1 A1 E2 04 04 04 <DE>
0A90:04 04 E2 A1 A1 A1 A1 A1 <67>
0A98:A1 A1 A1 A1 A1 A1 A1 A1 <51>
0AA0:A1 A1 A1 A1 A1 A1 A1 A1 <09>
0AA8:A1 A1 A1 A1 A1 A1 A1 A1 <01>
0AB0:A1 A1 A1 A1 A8 80 80 80 <6B>
0AB8:80 80 B2 A1 A1 A1 A1 A1 <6B>
0AC0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
0AC8:A1 A1 A1 A1 A1 A1 A1 A1 <E1>
0AD0:A1 A1 A1 A1 A1 A1 A1 A1 <F9>
0AD8:A1 A1 A1 A1 B8 80 80 80 <E2>
0AE0:80 80 B9 A1 A1 A1 A1 A1 <1F>
0AE8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0AF0:72 14 17 7C 00 00 00 00 <AC>
0AF8:00 00 00 00 00 00 00 00 <0C>
0B00:00 00 00 00 00 00 00 00 <16>
```

»Level 2« (Schluß)

```
Programmname :BILD02.DAT
Länge        :2816 Bytes
----------------------------------
0000:A1 A1 A1 A1 A8 E1 B2 A1 <1E>
0008:A1 A1 A1 A1 A1 A1 A1 A1 <BD>
0010:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0018:A1 A1 A1 A1 B6 B6 A1 A1 <AA>
0020:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0028:A1 A1 A1 A1 A9 E1 A2 A1 <1E>
0030:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0038:A1 A1 A8 A2 A1 A1 A1 A1 <75>
0040:B8 AD A1 A1 B6 B6 A1 A1 <EE>
0048:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0050:A1 A1 A1 A1 A3 E1 80 A4 <46>
0058:A1 A1 A1 A1 A3 AD AE AD <D3>
0060:A3 B8 AD 80 AD AD AE AD <5D>
0068:00 00 AD B9 B6 B6 B8 2F <D6>
0070:30 B9 A1 A1 A1 A1 A1 A1 <00>
0078:A1 A1 A3 AD 00 E1 00 A4 <7B>
0080:A1 A1 A1 A3 00 00 00 00 <F7>
0088:00 00 00 00 00 00 00 00 <88>
0090:00 00 00 00 8A 8B 00 00 <13>
0098:00 00 A2 A1 A1 A1 A1 A1 <42>
00A0:A1 A8 00 00 00 E1 00 A2 <80>
00A8:A1 A1 A3 00 00 00 00 00 <16>
00B0:00 00 00 00 00 00 00 00 <B0>
00B8:00 00 00 00 8A 8B 00 00 <3B>
00C0:00 00 A6 A1 A1 A1 A1 A1 <1D>
00C8:A1 A1 A7 00 00 E1 00 00 <B9>
00D0:A2 A3 00 00 00 00 00 00 <4A>
00D8:00 00 80 80 B4 B5 A7 80 <06>
00E0:00 00 00 00 8A 8B 00 00 <63>
00E8:00 00 B9 A1 A1 A1 A1 A1 <8B>
00F0:A1 A1 AD 00 00 E1 00 00 <E2>
00F8:00 00 00 00 00 00 00 00 <F8>
0100:00 00 B4 B5 A1 A1 A1 A7 <57>
0108:00 00 00 00 8A 8B 00 00 <88>
0110:00 00 B2 A1 A1 A1 A1 A1 <C4>
0118:A1 A1 A7 00 00 E1 00 00 <6B>
0120:00 00 00 00 00 00 00 80 <A2>
0128:B4 B5 A1 A1 A1 A1 A1 B8 <24>
0130:00 00 00 00 8A 8B 00 00 <B0>
0138:00 00 B2 A1 A1 A1 A1 A1 <FE>
0140:A1 A1 A1 A7 00 E1 00 00 <FC>
0148:00 00 00 00 00 00 B4 B5 <68>
0150:A1 A1 A1 A1 A1 A1 A1 A9 <84>
0158:00 00 00 00 8A 8B 00 00 <D8>
0160:00 00 B9 A1 A1 A1 A1 A1 <11>
0168:A1 A1 A1 A1 F0 E1 E1 E1 <C8>
0170:E1 E1 E1 E1 E1 E1 E1 F7 <8C>
0178:A1 A1 A1 A1 A1 A1 A1 A8 <CA>
0180:00 00 00 00 8A 8B 00 00 <01>
0188:00 00 B2 A1 A1 A1 A1 A1 <4E>
0190:A1 A1 A1 A1 F8 E1 E1 E1 <AF>
0198:E1 E1 E1 E1 E1 E1 E1 F7 <D2>
01A0:A1 A1 A1 A1 A1 A1 A1 A8 <32>
01A8:00 00 00 00 80 80 80 00 <AD>
01B0:00 00 A4 A1 A1 A1 A1 A1 <A3>
01B8:A1 A1 A1 A1 F1 E1 E1 E1 <3F>
01C0:E1 E1 F5 F3 F1 E1 E1 F7 <FD>
01C8:A1 A1 A1 A1 A1 A1 A1 A1 <F3>
01D0:E2 04 04 04 04 04 04 04 <C1>
01D8:04 E2 B2 A1 A1 A1 A1 A1 <FF>
01E0:A1 A1 A1 A1 F8 E1 E1 E1 <5F>
01E8:F7 F3 F5 A1 F8 F5 F3 F5 <4A>
01F0:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
01F8:A1 A7 00 00 00 00 00 00 <35>
0200:00 B2 A1 A1 A1 A1 A1 A1 <4A>
0208:A1 A1 A1 A1 F8 E1 E1 F3 <3B>
0210:F5 A1 A1 A1 A1 A1 A3 AD <90>
0218:B9 A1 A1 A1 A1 A1 A1 A1 <DD>
0220:A1 A1 B5 AA AA AB AA A7 <52>
0228:A6 A1 A1 A1 A1 A1 A1 A1 <4F>
0230:A1 A1 A1 A1 F8 E1 E1 F7 <17>
0238:A1 A1 B8 AE A3 AE 80 00 <93>
0240:80 AD AD A2 A1 A1 A1 A1 <9E>
0248:A1 A1 A1 A1 A1 A1 A1 A1 <72>
0250:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0258:A1 A1 A1 A1 F1 E1 E1 A2 <61>
0260:B3 A3 00 00 00 00 00 00 <66>
0268:00 00 00 00 A2 A1 A1 A1 <E2>
0270:A1 A1 A1 A3 AE AD B9 AD <BE>
0278:AE A2 A1 A1 A1 A1 A1 A1 <9B>
0280:A1 A1 A1 A1 F8 E1 E1 00 <F0>
0288:00 00 00 00 00 00 00 00 <8C>
0290:00 00 00 00 00 00 B9 A1 A1 <5D>
0298:A1 B8 A3 00 00 00 00 00 <AF>
02A0:00 00 63 B9 A1 A1 A1 A1 <18>
02A8:A1 A1 A1 A1 F1 E1 E1 E1 <91>
02B0:E1 E1 E1 E1 E1 E1 E1 E1 <B6>
02B8:F6 A7 00 00 00 B2 A1 B8 <23>
02C0:AE 00 00 00 00 00 00 00 <9B>
02C8:00 00 00 00 80 B2 A1 A1 <B7>
02D0:A1 A1 A1 A1 F8 E1 E1 E1 <61>
02D8:E1 E1 E1 E1 E1 E1 E1 E1 <ED>
02E0:F5 A1 A7 00 00 00 AE 00 <D9>
02E8:00 00 00 00 00 00 B4 AC <00>
02F0:80 00 00 00 80 B2 A1 A1 <5D>
02F8:A1 A1 A1 A1 F1 F2 F2 F1 <F8>
0300:F8 E1 F6 F1 F5 E1 E1 E1 <1E>
0308:F7 A1 A8 00 00 00 80 00 <E7>
0310:00 00 00 A6 AB B5 A1 A1 <84>
0318:AB AC 00 00 00 A2 A1 A1 <47>
0320:A1 A1 A1 A1 A1 A1 A1 A1 <B7>
0328:F0 F2 F5 A1 F8 E1 E1 E1 <41>
0330:F5 A1 A9 00 00 00 00 00 <ED>
0338:00 A6 AA A1 A1 A1 A1 A1 <76>
0340:A1 A1 A7 00 00 00 A4 A1 <BC>
0348:A1 A1 A1 A1 A1 A1 A1 A1 <70>
0350:A1 A1 A1 A1 F1 F3 F3 F3 <29>
0358:F5 A1 A1 AB AC B4 AC B4 <29>
0360:B5 A1 A1 A1 A1 A1 A1 A1 <D5>
0368:A1 A1 A8 00 00 00 A4 A1 <E4>
0370:A1 A1 A1 A1 A1 A1 A1 A1 <C7>
0378:A1 A1 A1 A1 A1 A1 A1 A1 <BF>
0380:A1 A1 A1 A1 A1 A1 A1 A1 <57>
0388:A1 A1 A1 A1 A1 A1 A1 A1 <2F>
0390:A1 A1 A3 00 00 00 B2 A1 <E8>
0398:A1 A1 A1 A1 A1 A1 A1 A1 <1F>
03A0:A1 A1 A1 A1 A1 A1 A1 A1 <37>
03A8:A1 A1 A1 A1 A1 A1 A1 A1 <4F>
03B0:A1 A1 A1 A1 A1 A1 A1 A1 <07>
03B8:A1 A1 A9 00 00 00 A2 A1 <72>
03C0:A1 A1 A1 A1 A1 A1 A1 A1 <17>
03C8:A1 A1 A1 A1 A1 A1 A1 A1 <EF>
03D0:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
03D8:A1 A1 A1 A1 A1 A1 A1 09 <47>
03E0:A1 A1 A8 00 00 00 B2 A1 <5A>
03E8:A1 A1 A1 A1 A1 A1 A1 A1 <0F>
03F0:A1 A1 A1 A1 A1 A1 A1 A1 <47>
03F8:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
0400:A1 A1 A1 A1 A1 A1 A3 6E <8E>
0408:07 A1 A9 00 00 00 A2 A1 <F5>
0410:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0418:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0420:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0428:A1 A3 B9 B8 A2 A3 6F A6 <2A>
0430:A1 A1 A3 00 00 00 B2 A1 <49>
0438:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0440:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0448:A1 A1 A1 A1 A1 A1 A1 A3 <78>
0450:AD 6E 6D 6C 6B 6A 00 A2 <1D>
0458:B9 A3 80 00 80 80 A4 A1 <5B>
0460:A1 A1 A1 A1 A1 A1 A1 A1 <9E>
0468:A1 A1 A1 A1 A1 A1 A1 A1 <D5>
0470:A1 A1 A1 A3 AE AD AE 6A <68>
0478:6F 80 80 80 80 80 80 80 <85>
0480:00 80 80 80 80 80 A2 A1 <9C>
0488:A1 A1 A1 A1 A1 A1 A1 A1 <35>
0490:A1 A1 A1 A1 A1 A1 A1 A1 <2D>
0498:A1 A1 A2 80 6D 6C 6B 80 <31>
04A0:80 80 80 80 80 80 80 80 <57>
04A8:80 80 80 80 80 B4 B5 A1 <AA>
04B0:A1 A1 A1 A1 A1 A1 A1 A1 <0D>
04B8:A1 A1 A1 A1 A1 A1 A1 A1 <05>
04C0:A1 A1 A7 6E 80 80 80 80 <9A>
04C8:80 80 80 80 80 80 80 80 <2F>
04D0:80 6F 6A 6B A2 A1 A1 A1 <D6>
04D8:A1 A1 A1 A1 A1 A1 A1 A1 <E5>
04E0:A1 A1 A1 A1 A1 A1 A1 A1 <1D>
04E8:A1 A1 A3 6F 80 80 80 80 <82>
04F0:80 80 80 80 80 80 80 6D <F4>
04F8:6E B4 B5 A7 6C A2 A1 A1 <0D>
0500:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0508:A1 A1 A1 A1 A1 A1 A1 A1 <BB>
0510:A1 A1 AD 6A 80 80 80 80 <CC>
0518:80 80 80 80 80 6B 6C B4 <94>
0520:B5 A1 A1 A1 A7 6D 07 A1 <B5>
0528:A1 A1 A1 A1 A1 A1 A1 A1 <9C>
0530:A1 A1 A1 A1 A1 A1 A1 A1 <84>
0538:A1 A7 6B 80 80 80 80 80 <46>
0540:80 80 80 6F 6A B4 B5 A1 <51>
0548:A1 A1 A1 A1 A1 08 A1 A1 <1A>
0550:A1 A1 A1 A1 A1 A1 A1 A1 <64>
0558:A1 A1 A1 A1 A1 A1 A1 A1 <6C>
0560:A1 B8 6C 80 80 80 80 80 <B9>
0568:80 6D 6E B4 B5 A1 A1 A1 <62>
0570:A1 A1 A1 A1 A1 A1 A1 A1 <C3>
0578:A1 A1 A1 A1 A1 A1 A1 A1 <4B>
0580:A1 A1 A1 A1 A1 A1 A1 A1 <33>
0588:A1 AA A7 6D 80 80 80 6B <E9>
0590:6C B4 B5 A1 A1 A1 A1 A1 <1A>
0598:A1 A1 A1 A1 A1 A1 A1 A1 <2B>
05A0:A1 A1 A1 A1 A1 A1 A1 A1 <53>
05A8:A1 A1 A1 A1 A1 A1 A1 A1 <1B>
05B0:A1 A1 A8 6E 80 80 6A B4 <D0>
05B8:B5 A1 A1 A1 A1 A1 A1 A1 <F1>
05C0:A1 A1 A1 A1 A1 A1 A1 A1 <F3>
05C8:A1 A1 A1 A1 A1 A1 A1 A1 <FB>
05D0:A1 A1 A1 A1 A1 A1 A1 A1 <E3>
05D8:A1 A1 A9 6F 80 6F B5 A1 <27>
05E0:A1 A1 A1 A1 A1 A1 A1 A1 <13>
05E8:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
05F0:A1 A1 A1 A1 A1 A1 A1 A1 <43>
05F8:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
0600:A1 A1 A8 6A 80 6E B2 A1 <74>
0608:A1 A1 A1 A1 A1 A1 A1 A1 <A9>
0610:A1 A1 A1 A1 A1 A1 A1 A1 <A1>
0618:A1 A1 A1 A1 A1 A1 A1 A1 <B9>
0620:A1 A1 A1 A1 A1 A1 A1 A1 <D1>
0628:A1 A1 B8 6B 80 6D A4 A1 <3A>
0630:A1 A1 A1 A1 A1 A1 A1 A1 <82>
0638:A1 A1 A1 A1 A1 A1 A1 A1 <9A>
0640:A1 A1 A1 A1 A1 A1 A1 A1 <72>
0648:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0650:A1 A1 A8 6C 80 6C A4 A1 <20>
0658:A1 A1 09 A1 A1 A1 A1 A1 <7D>
0660:A1 A1 A1 A1 A1 A1 A1 A1 <92>
0668:A1 A1 A1 A1 A1 A1 A1 A1 <C9>
0670:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0678:A1 A1 A8 6D 80 6B B9 A1 <4D>
0680:A1 A1 6A A1 A1 A1 A1 A1 <58>
0688:A1 A1 A1 A1 A1 A1 A1 A1 <29>
0690:A1 A1 A1 A1 A1 A1 A1 A1 <21>
0698:A1 A1 A1 A1 A1 A1 A1 A1 <39>
06A0:A1 A1 A3 6E 80 6A A2 B8 <9B>
06A8:B9 A3 6B A1 A1 A1 A1 A1 <C3>
06B0:A1 A1 A1 A1 A1 A1 A1 A1 <01>
06B8:A1 A1 A1 A1 A1 A1 A1 A1 <19>
06C0:A1 A1 A1 A1 A1 A1 A1 A1 <F1>
06C8:A1 A3 6F 80 80 80 6F 6E <00>
06D0:6D 6C 80 A2 A1 A1 A1 A1 <CE>
06D8:A1 A1 A1 A1 A1 A1 A1 A1 <F9>
06E0:A1 A1 A1 A1 A1 A1 A1 A1 <11>
06E8:A1 A1 A1 A1 A1 A1 A1 A1 <49>
06F0:A3 6A 80 80 80 80 80 80 <77>
06F8:80 80 80 80 A2 B8 63 AE <DD>
0700:B9 A1 A1 A1 A1 A1 A1 A1 <A3>
0708:A1 A1 A1 A1 A1 A1 A1 A1 <A7>
```

»Level 3« (Anfang)

```
0710:A1 A1 A1 A1 A1 06 6D 6C <99>
0718:6B 80 80 80 80 80 80 80 <64>
0720:80 80 80 80 80 80 80 80 <D1>
0728:80 A2 A1 A1 A1 A1 A1 A1 <78>
0730:A1 A1 A1 A1 A1 A1 A1 A1 <80>
0738:A1 A1 A1 A1 A1 A1 A1 A9 <A0>
0740:80 80 80 80 80 80 80 80 <B1>
0748:80 80 80 80 80 80 80 80 <A9>
0750:80 80 AE B9 A1 A1 A1 A1 <FD>
0758:A1 A1 A1 A1 A1 A1 A1 A1 <78>
0760:A1 A1 A1 A1 A1 A1 A1 A1 <90>
0768:B1 80 80 80 80 80 80 80 <71>
0770:80 80 80 80 80 B7 80 80 <5D>
0778:80 80 80 80 AD A2 A1 A1 <D1>
0780:A1 A1 A1 A1 A1 A1 A1 A1 <2F>
0788:A1 A1 A1 A1 A1 A1 A1 A1 <27>
0790:B8 80 80 80 80 80 80 80 <45>
0798:B4 B5 AA B5 A5 A1 AB AC <29>
07A0:80 80 80 80 80 00 B2 A1 <D7>
07A8:A1 A1 A1 A1 A1 A1 A1 A1 <07>
07B0:A1 A1 A1 A1 A1 A1 A1 A1 <FF>
07B8:A9 B1 80 80 80 80 80 A6 <FE>
07C0:A1 A1 A1 A1 A1 A1 A1 A1 <EF>
07C8:A7 B1 80 80 80 80 B9 A1 <5C>
07D0:A1 A1 A1 A1 A1 A1 A1 A1 <DF>
07D8:A1 A1 A1 A1 A1 A1 A1 A1 <F7>
07E0:A1 B8 B1 A5 B7 B4 B5 A1 <2E>
07E8:A1 A1 A1 A1 A1 A1 A1 A1 <47>
07F0:A1 A1 A7 00 80 80 B2 A1 <90>
07F8:A1 A1 A1 09 A1 A1 A1 A1 <51>
0800:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0808:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0810:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0818:A1 A1 A1 A7 80 80 B9 A1 <DC>
0820:A1 A1 A1 6B A1 A1 A1 A1 <31>
0828:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0830:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0838:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0840:A1 A1 A1 A3 80 80 A4 A1 <09>
0848:A1 A1 A1 6A A1 A1 A1 A1 <F9>
0850:A1 A1 A1 A8 B9 A1 A1 A1 <95>
0858:A1 A1 A1 A1 A1 A1 A1 A1 <9E>
0860:A1 A1 A1 A1 A1 A1 A1 A1 <D5>
0868:A1 A1 A3 00 80 80 B2 A1 <87>
0870:A1 A1 A1 6F A2 B8 B9 A3 <B5>
0878:AE B8 AE AD 00 AE A2 A3 <FB>
0880:A2 A3 AD AE B8 AE B9 A3 <97>
0888:A2 AD AE AE B8 AE B9 A3 <30>
0890:AD AD 80 00 80 80 B9 A1 <C7>
0898:A1 A1 A3 80 6E 80 80 80 <0A>
08A0:80 80 80 80 80 80 80 80 <4F>
08A8:80 80 80 80 80 80 80 80 <47>
08B0:80 80 80 80 80 80 80 80 <3F>
08B8:80 80 80 80 80 80 B2 A1 <BB>
08C0:A1 A8 00 80 00 6D 6C 6B <F8>
08C8:6A 6F 6E 6D 6C 6B 6A 6F <5A>
08D0:6E 6D 6C 6B 6A 6F 6E 6D <65>
08D8:6C 6B 6A 6F 6E 6D 6C 6B <02>
08E0:6A 6F 80 80 80 80 A4 A1 <47>
08E8:A1 B8 00 00 00 00 00 00 <77>
08F0:00 00 00 00 00 00 00 00 <00>
08F8:00 00 00 00 00 00 00 00 <08>
0900:00 00 00 00 00 00 00 00 <12>
0908:00 00 6E 6D 00 A6 A1 A1 <17>
0910:A1 A9 00 00 80 A6 AA A5 <AC>
0918:B5 AA AA AA A5 AA B5 AA <F2>
0920:A5 A5 00 B1 AA B5 A5 A5 <0F>
0928:AA B1 AA AA B5 AA B1 A5 <B2>
0930:B5 AA A1 A7 6C A2 A1 A1 <6D>
0938:A1 A8 00 00 00 B9 A1 A1 <4D>
0940:A1 A1 A1 A1 A1 A1 A1 A1 <7C>
0948:A1 A3 A6 A1 A1 A1 A1 A1 <83>
0950:A1 A1 A1 A1 A1 A1 A1 A1 <6C>
0958:A1 A1 A1 A1 A7 6B A1 A1 <EC>
0960:A1 A8 00 00 00 B2 A1 A1 <99>
0968:A1 A1 A1 A1 A1 A1 A1 A1 <C3>
0970:A1 A1 A1 A1 A1 A1 A1 A1 <4B>
0978:A1 A1 A1 A1 A1 A1 A1 A1 <33>
0980:A1 A1 A1 A1 A1 6A A1 A1 <60>
0988:A1 B8 00 00 00 B2 A1 A1 <04>
0990:A1 A1 A1 A1 A1 A1 A1 A1 <2B>
0998:A1 A1 A1 A1 A1 A1 A1 A1 <53>
09A0:A1 A1 A1 A1 A1 A1 A1 A1 <1B>
09A8:A1 A1 A1 A1 A1 08 A1 A1 <A1>
09B0:A1 A8 00 00 00 A2 A1 A1 <AA>
09B8:A1 A1 A1 A1 A1 A1 A1 A1 <F3>
09C0:A1 A1 A1 A1 A1 A1 A1 A1 <FB>
09C8:A1 A1 A1 A1 A1 A1 A1 A1 <E3>
09D0:A1 A1 A1 A1 A1 A1 A1 A1 <EB>
09D8:A1 A9 00 00 00 00 AE A2 <62>
09E0:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
09E8:A1 A1 A1 A1 A1 A1 A1 A1 <43>
09F0:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
09F8:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0A00:A1 A3 00 00 00 00 00 00 <8D>
0A08:AD AE A2 B8 A2 A1 A1 A1 <8F>
0A10:A1 A1 A1 A1 A1 A1 A1 A1 <B9>
0A18:A1 A1 A1 A1 A1 A1 A1 A1 <D1>
0A20:A1 A1 A1 A1 A1 A1 A1 A1 <8A>
0A28:A1 A7 00 00 00 00 00 00 <76>
0A30:00 00 00 00 00 AD AE A2 <F8>
0A38:B8 B9 A1 A1 A1 A1 A1 A1 <E2>
0A40:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0A48:A1 A1 A1 A1 A1 A1 A1 A1 <66>
0A50:A1 A1 AB AC 00 00 00 00 <FC>
0A58:00 00 00 00 00 00 00 00 <6C>
0A60:00 00 AD A2 A1 A1 A1 A1 <91>
0A68:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0A70:A1 A1 A1 A1 A1 A1 A1 A1 <59>
0A78:A1 A1 A1 A1 A5 AC 00 00 <9B>
0A80:00 00 00 00 00 00 00 00 <94>
0A88:00 00 00 00 B2 A1 A1 A1 <90>
0A90:A1 A1 A1 A1 A1 A1 A1 A1 <39>
0A98:A1 A1 A1 A1 A1 A1 A1 A1 <51>
0AA0:A1 A1 A1 A1 A1 A1 AB AC <28>
0AA8:00 00 00 00 00 00 00 00 <BC>
0AB0:00 00 00 00 A2 A1 A1 A1 <38>
0AB8:A1 A1 A1 A1 A1 A1 A1 A1 <F1>
0AC0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
0AC8:A1 A1 A1 A1 A1 A1 A1 A1 <E1>
0AD0:AB AC 80 80 80 80 80 80 <BA>
0AD8:80 80 00 00 00 B9 A1 A1 <55>
0AE0:A1 A1 A1 A1 A1 A1 A1 A1 <49>
0AE8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0AF0:E2 A5 3F 74 00 00 00 00 <AD>
0AF8:00 00 00 00 00 00 00 00 <0C>
0B00:00 00 00 00 00 00 00 00 <16>
```

»Level 3« (Schluß)

```
Programmname :BILD03.DAT
Länge        :2816 Bytes
-----------------------------------

0000:A6 AA AB AA A7 A6 A5 AA <BD>
0008:B1 A5 AC A5 A7 00 00 00 <27>
0010:00 00 00 A6 A7 11 B1 B5 <12>
0018:AB AA B1 AC A5 AA AB A5 <F7>
0020:B5 A5 A5 B1 AA AA A7 80 <A9>
0028:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0030:A1 A1 A1 A1 A1 A7 00 00 <CA>
0038:00 00 E7 B2 A1 A5 B2 A1 <D8>
0040:A1 A1 A1 A1 A1 A1 A1 A1 <86>
0048:A1 AD A1 A1 A1 A1 A1 A7 <79>
0050:21 21 21 A1 A1 A1 21 21 <46>
0058:21 21 21 21 21 21 A5 AA <E0>
0060:B1 A6 AA 21 21 21 A1 A1 <F9>
0068:A1 A1 21 21 21 21 23 2F <11>
0070:30 00 22 21 21 21 21 21 <4D>
0078:A1 A1 A1 A1 A1 AE B2 A1 <23>
0080:A1 A1 A1 A1 A1 A1 A1 A1 <45>
0088:A1 A1 A1 A1 A1 A1 A1 A1 <3D>
0090:A1 A1 A1 A1 A1 A3 80 80 <DA>
0098:80 80 80 A4 A1 A1 A1 A1 <92>
00A0:A1 A1 A1 A1 A3 80 AE A2 <CC>
00A8:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
00B0:A1 A1 A1 A1 A1 A1 A1 A1 <15>
00B8:A1 A1 A1 A1 A3 80 80 80 <37>
00C0:80 80 80 B2 A1 A1 A1 A1 <49>
00C8:A1 A1 A9 AD 80 80 80 80 <C6>
00D0:AE AD A2 A1 AD A1 A1 A1 <E3>
00D8:A3 E2 A2 A1 A1 A1 A1 A1 <74>
00E0:A1 A1 A1 A3 80 80 80 80 <10>
00E8:80 80 A6 A1 A1 A1 A1 A1 <D5>
00F0:A1 A1 AD 80 80 80 80 80 <D0>
00F8:80 80 80 AE 80 A2 A1 A1 <D0>
0100:A9 83 B2 A3 AE A4 A1 AE <0D>
0108:AD AD B8 80 80 80 80 B1 <BD>
0110:80 A6 A1 A1 A1 A1 A1 A1 <9B>
0118:21 23 00 00 00 00 00 00 <73>
0120:00 00 00 00 00 00 2E AE <2C>
0128:2D 03 2E 00 00 24 23 00 <1C>
0130:00 00 00 00 00 00 00 24 <56>
0138:2A 21 21 21 21 A1 21 21 <BC>
0140:21 29 00 00 00 00 00 00 <1D>
0148:00 00 00 00 00 00 00 00 <4A>
0150:00 03 00 00 00 2D 00 00 <C7>
0158:00 00 00 00 00 00 00 24 <7E>
0160:21 21 21 21 21 21 21 21 <52>
0168:21 28 00 00 00 00 00 31 <F5>
0170:26 2A 2B A5 00 00 00 00 <4F>
0178:00 03 00 00 00 00 00 00 <3B>
0180:00 00 00 00 00 00 00 32 <B4>
0188:21 21 21 2E 2D 21 21 21 <B7>
0190:21 21 2B 2C 00 00 00 22 <B5>
0198:21 33 21 A8 2C 00 00 00 <F7>
01A0:00 03 00 00 00 00 00 00 <63>
01A8:31 00 00 00 00 00 31 24 <C9>
01B0:21 21 23 6E 6D 22 21 21 <15>
01B8:21 23 AE 2D 00 00 00 00 <9C>
01C0:2D 00 22 21 21 2B AA A9 <59>
01C8:00 62 00 00 26 2A AA 31 <C1>
01D0:33 25 2A 27 26 25 21 21 <96>
01D8:21 28 6F 26 27 6C 24 21 <F5>
01E0:A9 80 80 80 80 80 80 80 <88>
01E8:80 80 80 AD AE AD A2 A3 <6B>
01F0:01 B2 AA A1 A1 A1 A1 A1 <9E>
01F8:A1 A1 A1 A1 A1 A1 A1 A1 <43>
0200:A1 A6 EA A4 A9 EB B2 A1 <6B>
0208:A1 AA A7 80 80 B1 80 80 <77>
0210:80 80 80 80 80 80 80 01 <6C>
0218:A6 AE A2 A1 A1 A1 A1 A1 <8C>
0220:A1 A1 A1 A1 A1 A3 B2 A1 <E3>
0228:AE A3 EB B2 A1 EA A1 A1 <F3>
0230:A1 A1 A1 A7 A6 A1 AB AC <30>
0238:80 80 80 80 80 80 01 80 <C4>
0240:AD 80 80 AE A2 A1 A1 A1 <F7>
0248:A1 A1 A3 AE AD 80 AE A3 <79>
0250:80 80 EC B2 A1 08 A1 A1 <E6>
0258:A1 A1 A1 A1 A1 A1 A1 A8 <6D>
0260:80 80 80 80 80 01 80 80 <9D>
0268:80 80 80 80 A6 A1 A1 B8 <C1>
0270:AE 63 80 80 80 80 80 80 <BB>
0278:80 80 ED 80 A2 A1 A1 A1 <2B>
0280:A1 A1 A1 A1 A1 A1 A1 A7 <5F>
0288:80 B1 80 80 01 80 80 80 <AA>
0290:80 80 80 B2 A1 A1 A8 80 <62>
0298:80 80 80 80 80 80 80 80 <63>
02A0:80 80 EE 80 80 AD A2 A1 <25>
02A8:A1 A1 A1 A1 A1 A1 A1 A1 <51>
02B0:A1 A8 00 01 00 80 80 80 <BC>
02B8:80 80 80 A2 AE A1 A9 80 <AC>
02C0:80 80 80 80 80 80 80 80 <3B>
02C8:80 80 EF 80 80 80 A6 A1 <72>
02D0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
02D8:A1 A9 E2 B1 B5 AB AC 80 <DA>
02E0:80 80 80 80 80 B2 A9 80 <34>
02E8:80 80 80 A6 AA A5 A1 A7 <C3>
02F0:80 80 EA 80 B1 A6 A1 A1 <E0>
02F8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0300:A1 A1 A1 A1 A1 A1 A1 AB <E1>
0308:AC 80 80 80 80 A2 A3 80 <A8>
0310:80 80 A6 A1 A1 A1 A1 A1 <A0>
0318:A7 80 EB 80 B2 A1 A1 A1 <1D>
0320:A1 A1 A1 A1 A1 A1 A1 A1 <B7>
0328:A1 A1 A1 A1 A1 A1 A1 A1 <CF>
0330:A9 A6 A7 80 80 80 80 80 <86>
0338:80 B2 A1 A1 A1 A1 A1 A1 <9C>
0340:A9 80 EC 80 A4 A1 A1 A1 <BE>
0348:A1 A1 A1 A1 A1 A1 A1 A1 <70>
0350:A1 A1 A1 A1 A1 A1 A1 A1 <68>
0358:A1 F8 E1 E1 E1 E1 E1 E1 <04>
0360:E1 E1 F7 A1 A1 A1 A1 A1 <65>
0368:A7 80 ED 80 A6 A1 A1 A1 <2D>
0370:A1 A1 A1 A1 A1 A1 A1 A1 <C7>
0378:A1 A1 A1 A1 A1 A1 A1 A1 <BF>
0380:A1 F8 E1 E1 E1 E1 E1 E1 <BE>
0388:E1 E1 F7 A1 A1 A1 A1 A1 <14>
0390:A9 80 EE 80 B2 A1 A1 A1 <7C>
0398:A1 A1 A1 A1 A1 A1 A1 A1 <1F>
03A0:A1 A1 A1 A1 A1 A1 A1 A1 <37>
03A8:A1 F1 F3 F1 F4 F3 F1 F0 <66>
03B0:F7 F3 F5 A1 A1 A1 A1 A1 <05>
03B8:A9 00 EF 80 A2 A1 A1 A1 <74>
03C0:A1 A1 A1 A1 A1 A1 A1 A1 <17>
03C8:A1 A1 A1 A1 A1 A1 A1 A1 <EF>
03D0:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
03D8:A1 A1 A1 A1 A1 A1 A1 A1 <DF>
03E0:A3 00 EA 80 A4 A1 A1 A1 <FF>
03E8:A1 A1 A1 A1 A1 A8 AD AE <50>
03F0:A2 A3 AD AE AE 09 A1 A1 <9E>
03F8:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
0400:A1 A1 A1 A1 A1 A1 A1 A1 <BD>
0408:A9 00 EB 80 B2 A1 A1 A1 <46>
0410:A1 A1 A1 A1 A1 A3 00 00 <D2>
0418:00 00 00 00 00 EC 00 A2 <75>
0420:A1 A1 A1 A3 A2 A3 AD A2 <26>
0428:AD A4 A1 A1 A1 A1 A1 A1 <47>
0430:A3 80 EC 80 A2 A1 A1 A1 <A5>
0438:A1 A1 A1 A9 AD 00 00 00 <FC>
0440:00 00 00 00 EB 00 ED 00 <81>
0448:A2 A3 AD 00 00 00 00 00 <7F>
0450:B4 A1 A1 A1 A1 A1 A3 AE <FD>
0458:80 80 ED 80 A6 A1 A1 A1 <47>
0460:A1 A1 A1 A3 63 00 00 00 <62>
0468:00 00 00 EA 00 00 00 EE <0D>
0470:00 00 00 00 00 00 00 00 <78>
0478:B2 A1 A1 A1 A1 A3 00 80 <5B>
0480:80 80 EE 80 B2 A1 A1 A1 <A5>
0488:A1 A1 B8 00 00 00 00 00 <80>
```

```
0490:00 00 EF 00 00 00 00 00 <76>
0498:EF 00 00 00 00 00 00 00 <18>
04A0:A2 A1 A1 A1 A3 00 00 00 <7D>
04A8:80 80 EF 80 A6 A1 A1 A1 <3A>
04B0:A1 A9 00 00 00 00 00 00 <73>
04B8:00 EE A6 A7 B1 00 00 00 <31>
04C0:00 EA 00 00 00 00 00 00 <43>
04C8:00 B9 AD A3 00 00 00 A6 <B5>
04D0:A7 80 EA 80 A4 A1 A1 A1 <CB>
04D8:A1 A3 00 00 00 00 00 00 <D9>
04E0:ED 00 B2 A1 F8 E1 E1 E1 <95>
04E8:F6 AC EB 00 00 00 00 00 <34>
04F0:00 00 80 00 00 00 A6 A1 <D5>
04F8:A8 80 EB 80 A2 A1 A1 A1 <53>
0500:A1 A9 B1 00 00 00 00 EC <27>
0508:00 B2 A1 A1 F1 F2 F5 F4 <1A>
0510:A3 AD 00 EC 00 A6 AC B7 <7C>
0518:00 00 00 00 00 A6 A1 A1 <9C>
0520:A9 80 EC 80 80 A4 A1 A1 <C5>
0528:A1 A1 A3 00 00 00 EB 00 <35>
0530:00 A6 A1 A1 A1 A1 A1 A9 <5B>
0538:00 00 00 00 ED A2 A1 A1 <14>
0540:A7 00 00 00 A6 A1 A1 A1 <2F>
0548:A9 80 ED 80 A6 A1 A1 A1 <C2>
0550:A1 A3 00 00 00 EA 00 00 <FA>
0558:A6 A1 A9 AE B2 A1 A1 A1 <4F>
0560:AC 00 00 00 A7 EE 07 A1 <E4>
0568:A1 A5 B1 A1 A1 A1 A1 A1 <CA>
0570:A3 80 EE 80 A4 A1 A1 A1 <2B>
0578:A1 A9 00 00 EF 00 00 00 <35>
0580:B2 A1 A7 A6 A1 A1 A7 A2 <C6>
0588:A3 00 00 00 B2 08 A1 A1 <F6>
0590:A1 A1 A1 A1 A1 A1 A1 A1 <23>
0598:A8 80 EF 80 B2 A1 A1 A1 <32>
05A0:A1 A9 00 EE 00 00 00 A6 <2A>
05A8:A1 A1 A1 A1 A1 A3 AD 00 <9A>
05B0:00 00 80 00 B2 A1 A1 A1 <A1>
05B8:A1 A1 A1 A1 A1 AD A1 A1 <3B>
05C0:A8 80 EA 80 A4 A1 A1 A1 <79>
05C8:A1 06 ED 00 00 00 A6 A1 <30>
05D0:A1 A1 A1 A1 B8 00 80 80 <B6>
05D8:00 00 80 80 AD A2 A1 A1 <CD>
05E0:A1 A1 A1 A1 A1 AA A1 A1 <37>
05E8:A8 80 EB 80 AD A4 A1 A1 <24>
05F0:A1 A9 00 EC 00 00 AD AE <7D>
05F8:A2 A1 A1 A1 A3 80 80 80 <75>
0600:80 80 80 80 80 80 B2 A1 <78>
0608:A1 A1 A1 A1 A1 A1 A1 A1 <A9>
0610:A3 80 EC 80 80 A4 A1 A1 <B8>
0618:A1 A3 00 00 EB 00 00 00 <F4>
0620:A6 A1 A1 A9 80 80 80 80 <E1>
0628:B1 80 80 80 80 80 A2 A1 <98>
0630:A1 A1 A1 A1 A1 A1 A1 A3 <84>
0638:80 80 ED 80 A6 A1 A1 A1 <63>
0640:A1 A9 00 00 00 EA 00 00 <6E>
0648:A2 A1 A1 A1 A7 80 80 80 <33>
0650:A4 AB B1 80 80 80 80 E2 <C3>
0658:A1 A1 A1 A1 A1 A1 E2 00 <5A>
0660:80 80 EE 80 A2 A1 A1 A1 <3B>
0668:A1 A9 00 00 00 00 EF 00 <0D>
0670:A6 A1 A1 A1 A8 80 80 80 <90>
0678:B2 A1 A1 A7 80 80 81 83 <28>
0680:A2 A1 A1 A1 A1 A3 83 80 <5C>
0688:80 80 EF B1 A6 A1 A1 A1 <7F>
0690:A1 A3 00 00 00 00 00 EE <04>
0698:B2 09 09 A1 A3 80 80 80 <D3>
06A0:A2 A1 A1 A3 80 81 80 83 <0A>
06A8:80 B2 A1 A1 A3 80 83 80 <4D>
06B0:80 B1 EA B2 A1 A1 A1 A1 <F3>
06B8:A1 A9 00 00 00 00 00 00 <7F>
06C0:ED EC EB EA EF EE ED EC <6F>
06C8:EB EA 07 A7 81 80 80 83 <A2>
06D0:A6 A1 A1 A1 AB AC 83 80 <7E>
06D8:80 AD 08 A1 A1 A1 A1 A1 <5E>
06E0:A1 A1 A7 00 00 00 00 EE <48>
06E8:A6 08 08 AB 80 80 80 A6 <18>
06F0:A1 A1 A3 81 80 80 80 83 <98>
06F8:B9 A1 A1 A3 80 AD 83 80 <9D>
0700:80 80 B2 A1 A1 A1 A1 A1 <2A>
0708:A1 A9 AD 00 00 00 EF 00 <84>
0710:B2 A1 A1 A3 80 80 A6 A1 <D8>
0718:A1 A3 81 B1 80 80 80 83 <40>
0720:80 A2 B8 80 80 80 83 80 <48>
0728:80 A6 A1 A1 A1 A1 A1 A1 <77>
0730:A1 A9 00 00 00 EA 00 A6 <46>
0738:A1 A1 A9 A6 80 80 AD A2 <75>
0740:A9 E2 A6 A9 80 80 80 83 <E0>
0748:80 80 80 80 80 80 83 80 <AF>
0750:A6 A1 A1 A1 A1 A1 A1 A1 <E5>
0758:A1 A8 B1 00 EB 00 00 AD <DB>
0760:B2 A1 A1 A9 80 80 80 A6 <FF>
0768:A1 A1 09 A1 A5 AC 80 83 <CA>
0770:80 80 80 80 80 80 83 80 <87>
0778:AE B2 A1 A1 A1 A1 A1 A1 <FC>
0780:A1 A1 06 EC 00 00 00 00 <07>
0788:A2 A1 A1 A9 80 80 80 A2 <5B>
0790:A1 A1 EF A1 A1 A8 80 83 <B9>
0798:80 80 80 A6 AA A7 E2 B1 <95>
07A0:A6 A1 A1 A1 A1 A1 A1 A1 <9D>
07A8:A1 A1 A3 ED 00 00 00 00 <92>
07B0:B2 A1 A1 A3 80 80 80 80 <C9>
07B8:A2 A1 EE A1 A1 A1 A7 E2 <82>
07C0:A6 AA AA A1 A1 A1 A1 A1 <C9>
07C8:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
07D0:A1 A9 00 00 EE 00 00 00 <C8>
07D8:A2 A1 A8 B1 80 80 80 80 <68>
07E0:80 AE ED A2 A1 A1 A1 A1 <43>
07E8:A1 09 AD AE A2 A3 B2 A1 <BE>
07F0:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
07F8:A1 A9 00 00 00 EF 00 00 <BC>
0800:A6 A1 A1 A1 A7 80 80 80 <84>
0808:80 80 EC 80 A2 A1 A3 AE <80>
0810:AD EB 80 80 00 80 A2 A3 <91>
0818:AD AE AD AD AD A1 A1 A1 <A9>
0820:A1 A3 00 00 00 00 EA 00 <3D>
0828:B2 A1 A1 A1 A1 AB AC 80 <22>
0830:80 80 80 EB 80 AD 80 80 <1C>
0838:EC 80 80 00 00 00 00 00 <0E>
0840:00 00 80 B1 A6 A1 A1 A1 <F1>
0848:06 EB EA EF EE ED EC EB <28>
0850:07 A1 A1 A1 A1 A1 A1 AB <B6>
0858:AC 80 80 80 EA EF EE ED <D9>
0860:80 80 80 00 00 00 00 00 <00>
0868:00 00 A6 A1 A1 A1 A1 A1 <A5>
0870:A1 A5 00 00 00 00 00 00 <BA>
0878:B2 A1 A1 A1 A1 A1 A1 F8 <15>
0880:E1 E1 E1 E1 E1 E1 E1 E1 <D2>
0888:E1 E1 E1 F6 AB 00 00 00 <32>
0890:00 A6 A1 A1 A1 A1 A1 A1 <BE>
0898:A1 A3 00 00 00 00 00 00 <22>
08A0:A2 AD AD A3 AE A2 A1 F8 <EB>
08A8:E1 E1 E1 E1 E1 E1 E1 E1 <DA>
08B0:E1 E1 E1 F7 A9 F9 00 00 <1E>
08B8:00 A2 A1 A1 A1 A1 A1 A1 <0D>
08C0:A9 E2 04 04 04 04 04 04 <9A>
08C8:04 04 04 04 E2 00 B2 F1 <00>
08D0:F1 F7 F4 F2 F2 F5 F3 F2 <F1>
08D8:F5 F2 F3 F5 A3 00 00 00 <A2>
08E0:00 00 B2 A1 A1 A1 A1 A1 <A3>
08E8:A1 A9 00 00 00 00 00 00 <B3>
08F0:A6 AA AA A7 A1 A6 A1 A1 <2B>
08F8:A1 A1 A1 A3 AE A2 A3 AD <59>
0900:A2 A3 AE AD 00 00 00 00 <9C>
0908:F9 00 A4 A1 A1 A1 A1 A1 <80>
0910:A1 A1 A7 00 00 00 00 00 <EF>
0918:B2 A1 A1 A1 A1 A1 A1 A1 <1B>
0920:A1 A3 AD 00 00 00 80 00 <00>
0928:00 00 FA 00 00 00 00 00 <79>
0930:00 A6 A1 A1 A1 A1 A1 A1 <8B>
0938:A1 A1 A3 00 00 00 00 00 <37>
0940:B9 A1 A1 A1 A1 A1 A1 A1 <60>
0948:A9 00 00 00 00 00 F9 00 <A0>
0950:00 00 00 00 00 F9 00 00 <46>
0958:00 A2 A8 A2 A3 A2 A1 A1 <67>
0960:A1 A9 00 00 00 00 00 00 <2D>
0968:00 A2 A1 A1 A1 A1 A1 A1 <94>
0970:B8 00 F9 00 00 00 00 00 <FD>
0978:00 00 00 00 00 FA 00 00 <72>
0980:00 00 AD 00 FA 00 A6 A1 <EC>
0988:A1 A3 00 00 00 00 00 00 <14>
0990:00 00 B9 A1 A1 A1 A1 A3 <53>
0998:00 00 00 00 00 00 00 00 <AA>
09A0:A6 A7 F9 00 80 00 F9 00 <7C>
09A8:00 00 00 00 00 A6 A1 A1 <33>
09B0:A1 A7 00 00 00 00 00 00 <FC>
09B8:00 00 00 B9 A1 A1 A9 00 <2F>
09C0:00 00 00 F9 00 FA 00 A6 <F0>
09C8:B3 A3 00 00 00 FA 00 00 <C4>
09D0:00 00 00 00 B1 AD A2 A1 <03>
09D8:A1 A8 00 00 00 00 00 00 <25>
09E0:00 00 00 00 A2 A3 00 00 <8E>
09E8:00 00 00 00 00 00 00 AD <A7>
09F0:00 00 FA 00 00 00 00 00 <41>
09F8:00 00 00 00 A2 A7 A6 A1 <A6>
0A00:A1 A9 00 00 00 00 00 00 <0F>
0A08:00 00 00 00 00 00 00 00 <1C>
0A10:FA 80 00 00 00 00 F9 00 <F2>
0A18:00 F9 00 00 00 00 00 F9 <63>
0A20:00 00 00 FA 00 A2 A1 A1 <3E>
0A28:A1 A3 00 00 00 00 00 00 <75>
0A30:00 00 00 00 00 00 00 F9 <3D>
0A38:00 00 F9 00 FA 00 00 00 <3B>
0A40:00 00 00 00 00 00 FA 00 <48>
0A48:00 00 00 00 00 A6 A1 A1 <D9>
0A50:A9 00 00 00 00 00 00 00 <B8>
0A58:00 00 00 00 00 B4 AC 00 <96>
0A60:00 00 00 F9 00 F9 00 00 <EB>
0A68:FA 00 A6 A5 A6 AA A6 B5 <56>
0A70:A5 B1 A6 A7 A6 A1 A1 A1 <5E>
0A78:A9 00 00 00 00 00 00 00 <61>
0A80:00 00 00 B4 B5 A1 A1 AB <F6>
0A88:B5 AB AC B4 AC 00 00 00 <50>
0A90:00 A6 21 21 A1 A1 A1 21 <32>
0A98:21 21 21 21 21 A1 21 A1 <EE>
0AA0:A1 AB 00 00 00 00 00 00 <EF>
0AA8:80 B4 B5 A1 A1 A1 A1 A1 <0E>
0AB0:A1 A1 A1 A1 A1 AB B5 AB <73>
0AB8:A5 A1 A1 A1 A1 A1 A1 A1 <EF>
0AC0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
0AC8:A1 A1 AB AC 00 E8 80 B4 <B8>
0AD0:B5 A1 A1 A1 A1 A1 A1 A1 <0F>
0AD8:A1 A1 A1 A1 A1 A1 A1 A1 <11>
0AE0:A1 A1 A1 A1 A1 A1 A1 A1 <49>
0AE8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0AF0:12 E8 0C 74 00 00 00 00 <CF>
0AF8:00 00 00 00 00 00 00 00 <0C>
0B00:00 00 00 00 00 00 00 00 <16>
```

»Level 4«

```
Programmname :BILD04.DAT
Länge        :2816 Bytes
-----------------------------------

0000:8C 8C 8C 8C 8C 8C A1 A1 <26>
0008:A1 A1 A1 A1 A1 A1 A1 A1 <BD>
0010:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0018:A1 A1 A1 A1 A1 A1 A1 A1 <AD>
0020:A1 A1 A1 A1 A1 A1 8C 8C <67>
0028:8C 8C 8C 8C 8C 8C A1 A1 <02>
0030:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0038:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0040:A1 A1 A1 A1 A1 A3 A2 A1 <90>
0048:A1 A1 A1 A1 A1 A1 8C 8C <3F>
0050:8C 8C 8C 8C 8C 8C A1 A1 <D8>
0058:A1 A1 A1 A1 A1 A1 A1 A1 <6E>
0060:A1 A1 A1 A1 A1 A1 A1 A1 <62>
0068:A1 A1 A1 A1 A3 80 00 2F <75>
0070:30 00 B9 A1 A1 A1 8C 8C <7D>
0078:8C 8C 8C 8C 8C 8C A1 A1 <F2>
0080:A1 A1 A1 A1 A1 A1 A1 A1 <45>
0088:A1 A1 A1 A1 A1 A1 A1 A1 <3D>
0090:A1 A1 A1 A1 A7 00 00 00 <FC>
0098:00 00 00 A2 A1 A1 8C 8C <E7>
00A0:8C 8C 8C 8C 8C 8C A1 A1 <86>
00A8:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
00B0:A1 A1 A1 A1 A1 A1 A1 A1 <15>
00B8:A1 A1 A1 A1 B8 00 00 00 <5D>
00C0:00 00 00 A6 A1 A1 8C 8C <5B>
00C8:A1 8C 8C 8C.8C 8C A1 A1 <D3>
00D0:A1 A1 A1 A1 A1 A1 A1 A1 <F5>
00D8:A1 A1 A1 A1 A1 A1 A1 A1 <ED>
00E0:A1 A1 A1 A3 00 00 00 00 <97>
00E8:00 00 A6 A1 A1 A1 8C 8C <F6>
00F0:A1 8C 8C 8C 8C 8C A1 A1 <2B>
00F8:A1 A1 A1 A1 A1 A1 A1 A1 <4D>
0100:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
0108:A1 A1 B8 00 00 00 80 A6 <C1>
0110:A7 A6 A1 A1 A1 A1 8C 8C <A8>
0118:A1 8C 8C 8C 8C 8C A1 A1 <92>
0120:A1 A1 A1 A1 A1 A1 A1 A1 <AB>
0128:A1 A1 A1 A1 A1 A1 A1 A1 <D3>
0130:A1 A3 00 00 00 80 A6 A1 <55>
0138:A1 A1 A1 A1 A1 A1 8C 8C <45>
0140:A1 8C 8C 8C 8C 8C A1 A1 <62>
0148:A1 A1 A1 A1 A1 A1 A1 A1 <74>
0150:A1 A1 A1 A1 A1 A1 A1 A1 <7C>
0158:A3 00 00 00 80 A6 A1 A1 <2A>
0160:A1 A1 A1 A1 A1 A1 8C 8C <2D>
0168:A1 8C 8C 80 63 A2 A1 A1 <84>
0170:A1 A1 A1 A1 A1 A1 A1 A1 <DB>
0178:A1 A1 A1 A1 A1 A1 A1 B8 <DA>
0180:00 00 00 80 A6 A1 A1 A1 <26>
0188:A1 A1 A1 A1 A1 A1 8C 8C <F4>
0190:A1 8C 80 00 80 80 A2 A1 <36>
0198:A1 A1 A1 A1 A1 A1 A1 A1 <23>
01A0:A1 A1 A1 A1 A1 A1 A3 00 <8E>
01A8:00 00 80 A6 A1 A1 A1 A1 <80>
01B0:A1 A1 A1 A1 A1 A1 8C 8C <DC>
01B8:A1 8C 80 00 80 80 80 A2 <DB>
01C0:A1 A1 A1 A1 A1 A1 A1 A1 <0B>
01C8:A1 A1 A1 A1 A1 A3 80 00 <18>
01D0:00 00 A6 A1 A1 A1 A1 A1 <03>
01D8:A1 A1 A1 A1 A1 A1 8C 8C <A4>
01E0:A1 8C 80 00 80 80 80 80 <81>
01E8:A2 A1 A1 A1 A1 A1 A1 A1 <93>
01F0:A1 A1 A1 A1 A1 A7 80 00 <90>
01F8:00 00 A2 A1 A1 A1 A1 A1 <AB>
```

»Level 5« (Anfang)

```
0200:A1 A1 A1 A1 A1 A1 8C 8C <9A>
0208:A1 8C 80 00 80 80 80 80 <68>
0210:80 A2 A1 A1 A1 A1 A1 A1 <9A>
0218:A1 A1 A1 A1 A1 A1 A7 80 <8C>
0220:00 00 00 A2 A1 A1 A1 A1 <B8>
0228:A1 A1 A1 A1 A1 A1 8C 8C <92>
0230:A1 8C 80 00 A6 A7 80 80 <02>
0238:80 80 A2 A1 A1 A1 A1 A1 <1E>
0240:A1 A1 A1 A1 A1 A1 A1 A7 <A0>
0248:80 00 80 80 B2 A1 A1 A1 <08>
0250:A1 A1 A1 A1 A1 A1 8C 8C <2B>
0258:A1 E2 00 00 A2 A1 A7 80 <82>
0260:80 80 80 A2 A1 A1 A1 A1 <A8>
0268:A1 A1 A1 A1 A1 A1 A1 A8 <99>
0270:80 00 00 00 A2 A1 A1 A1 <AA>
0278:A1 A1 A1 A1 A1 A1 8C 8C <83>
0280:A1 8C 02 00 80 80 A1 A7 <A7>
0288:80 80 80 80 A2 A1 A1 A1 <68>
0290:A1 A1 A1 A1 A1 A1 A1 B8 <40>
0298:00 6B 6C 6B 80 A2 A1 A1 <0A>
02A0:A1 A1 A1 A1 A1 A1 8C 8C <FA>
02A8:A1 8C 80 02 80 00 A2 A1 <7A>
02B0:A7 80 80 80 80 A2 A1 A1 <A1>
02B8:A1 A1 A1 A3 A2 A1 A3 00 <8C>
02C0:6A 00 00 00 6A 00 A2 A1 <31>
02C8:A1 A1 A1 A1 A1 A1 8C 8C <B2>
02D0:A1 8C 00 00 02 80 80 A2 <BD>
02D8:A1 A7 80 80 80 00 A2 A1 <AF>
02E0:A1 A1 A3 80 00 AD 00 6F <DB>
02E8:00 00 00 00 00 6F 00 A2 <4C>
02F0:A1 A1 A3 A2 A1 A1 8C 8C <5A>
02F8:A1 8C 8C 8C 00 02 80 80 <33>
0300:A2 A1 A7 80 80 00 00 A2 <3A>
0308:A1 A3 00 00 00 00 6E 00 <62>
0310:00 80 80 80 00 00 6E 00 <0B>
0318:AD 63 00 A6 A1 A1 8C 8C <16>
0320:A1 8C E2 E2 80 80 02 00 <36>
0328:00 A2 A1 A7 80 80 00 00 <13>
0330:AD 00 00 80 00 00 6D 00 <5F>
0338:80 6B 80 6B 80 80 6D 00 <92>
0340:00 00 00 A2 A1 A1 8C 8C <97>
0348:A1 8C 03 03 02 00 00 02 <24>
0350:00 00 A2 A1 A7 80 80 00 <CA>
0358:00 00 00 00 00 00 6C 80 <B7>
0360:6C 80 6A 80 6C 00 6C 00 <9D>
0368:00 00 00 00 B9 A1 A1 A1 <9E>
0370:A1 8C 03 03 00 02 00 00 <82>
0378:02 00 00 A2 A1 A7 80 00 <40>
0380:00 00 00 00 00 00 6B 00 <5D>
0388:6D 00 6F 00 6D 00 6B 00 <75>
0390:00 00 00 00 00 A2 A1 A1 <FF>
0398:A1 8C 03 03 00 00 02 00 <E6>
03A0:00 02 01 00 A1 A1 A7 80 <A0>
03A8:00 00 00 00 00 00 6A 00 <83>
03B0:6E 00 6E 00 6E 00 6A 00 <02>
03B8:00 00 00 00 00 00 A2 A1 <A3>
03C0:A1 8C 03 03 00 00 00 02 <CC>
03C8:00 01 02 00 E2 A2 A1 A7 <CD>
03D0:80 80 00 00 00 00 6F 00 <95>
03D8:6F 00 6D 00 6F 00 6F 00 <FD>
03E0:00 A6 A7 00 00 00 00 A1 <C6>
03E8:A1 E2 03 03 00 00 00 00 <48>
03F0:01 00 00 02 03 00 A2 A1 <93>
03F8:A7 80 00 00 00 00 6E 00 <4F>
0400:6A 00 6C 00 6A 00 6E 00 <D8>
0408:A6 A1 A1 A7 00 00 00 A1 <CB>
0410:A1 8C 02 03 00 00 00 01 <3D>
0418:00 02 00 00 03 00 00 A2 <5A>
0420:A1 A7 80 00 00 00 6D 00 <8D>
0428:6B 00 6B 00 6B 00 6D 00 <89>
0430:A2 A1 A1 A1 A7 00 00 A1 <75>
0438:A1 8C 03 03 00 00 01 80 <C6>
0440:80 80 02 80 03 02 80 80 <01>
0448:A2 A1 A7 80 80 80 6C 00 <09>
0450:6C 00 6A 80 6C 80 6C 80 <0D>
0458:80 B9 A1 A1 A3 80 00 A1 <34>
0460:A1 8C 03 03 02 01 80 8C <CD>
0468:80 8C 80 02 03 80 02 80 <BD>
0470:80 A2 A1 A7 80 80 6B 00 <1E>
0478:6D A6 08 A7 6D 80 6B 80 <DA>
0480:80 A6 A1 A3 80 80 00 A1 <57>
0488:A1 8C 03 03 01 02 80 8C <71>
0490:80 8C 80 80 03 80 80 02 <FA>
0498:80 80 B2 A1 A7 80 6A 00 <31>
04A0:08 A1 A1 A1 08 80 6A 80 <84>
04A8:A6 A1 A3 80 80 80 A6 A1 <3C>
04B0:A1 8C 03 01 00 00 02 80 <20>
04B8:80 80 80 80 03 02 80 80 <59>
04C0:02 80 A4 A1 A1 A7 6F A6 <B5>
04C8:A1 A3 AD A2 A1 A7 6F A6 <C4>
04D0:A1 A1 A7 80 80 80 A2 A1 <0E>
04D8:A1 8C E2 04 04 04 04 02 <CA>
04E0:04 04 04 04 03 04 04 04 <E0>
04E8:04 E2 B2 A1 A1 A1 08 A1 <F8>
04F0:A3 80 80 80 A2 A1 08 A1 <BB>
04F8:A1 A1 A1 A7 80 80 00 A1 <52>
0500:A1 8C 03 03 00 00 80 80 <0F>
0508:02 80 80 80 03 80 80 02 <42>
0510:00 A2 A1 A1 8C 8C 09 A1 <0A>
0518:A7 80 80 80 A6 A1 09 A1 <30>
0520:A1 A1 A1 A3 80 80 00 A1 <28>
0528:A1 8C 03 03 00 00 00 80 <B6>
0530:80 02 80 80 03 80 01 80 <7A>
0538:02 80 B2 A1 8C 80 6C A2 <9C>
0540:A1 A7 B1 A6 A1 A3 6B A2 <DD>
0548:A1 A3 AD 80 80 80 00 A1 <8F>
0550:A1 8C 03 03 8C 8C 8C 8C <4D>
0558:00 80 02 80 03 01 80 8C <33>
0560:80 02 A4 A1 8C 80 6D 00 <88>
0568:A2 A1 A1 A1 A3 80 6A 80 <59>
0570:E2 04 04 04 04 04 E2 A1 <C2>
0578:A1 8C 03 03 8C 80 80 80 <E0>
0580:80 80 80 02 03 00 00 80 <B3>
0588:80 80 E2 A1 8C 80 6E 80 <FA>
0590:80 A2 09 A3 80 80 6F 80 <F7>
0598:AD 80 80 80 00 00 8C 8C <E4>
05A0:A1 8C 03 03 E2 04 04 04 <1A>
05A8:04 04 04 01 02 04 04 04 <72>
05B0:04 04 04 04 E2 80 6F 80 <EC>
05B8:80 80 6B 80 80 80 6E 80 <7A>
05C0:80 80 80 80 00 80 8C 8C <55>
05C8:A1 8C 03 03 03 02 80 80 <37>
05D0:80 80 01 00 03 02 80 80 <7B>
05D8:80 80 80 01 63 80 6A 80 <12>
05E0:80 80 6C 80 80 80 80 6D <84>
05E8:80 80 80 00 00 80 8C 8C <25>
05F0:A1 8C 03 03 03 00 02 80 <9B>
05F8:80 01 8C 8C 03 80 02 80 <7A>
0600:80 80 01 80 00 80 6B 80 <DD>
0608:80 80 6D 80 80 80 80 80 <8D>
0610:6C 80 80 00 80 80 8C 8C <09>
0618:A1 8C 03 03 03 00 80 02 <83>
0620:01 00 80 63 03 80 80 02 <FB>
0628:80 01 80 80 00 80 80 6C <1B>
0630:80 80 6E 80 80 80 80 80 <85>
0638:80 6B 80 00 80 80 8C 8C <6D>
0640:A1 8C 03 03 03 00 80 01 <4A>
0648:02 80 80 80 03 80 80 80 <06>
0650:02 80 80 A6 A5 80 80 80 <6F>
0658:6D 6E 6F 6A 6B 6C 6D 6E <FE>
0660:80 80 6A 00 80 80 8C 8C <02>
0668:A1 8C 03 03 03 00 01 00 <92>
0670:80 02 80 80 03 80 80 01 <C0>
0678:80 E2 A6 A1 8C 80 80 80 <2B>
0680:80 80 6E 80 80 80 80 80 <35>
0688:6F 80 80 6F 80 80 8C 8C <08>
0690:A1 8C 03 03 03 01 00 80 <7C>
0698:80 80 02 80 03 80 01 80 <A9>
06A0:80 A1 A1 A1 8C 80 80 80 <14>
06A8:80 80 6D 80 80 80 80 80 <ED>
06B0:80 6A 80 6A 6E 80 8C 8C <FE>
06B8:A1 8C 03 03 01 00 80 80 <51>
06C0:80 80 80 02 03 01 80 80 <79>
06C8:A6 A1 A1 A1 8C A7 80 80 <75>
06D0:80 80 6C 80 80 80 80 80 <A5>
06D8:80 80 6B 00 6D 80 8C 8C <02>
06E0:A1 8C 03 E2 03 E2 04 04 <5A>
06E8:04 04 04 04 E2 80 80 A6 <72>
06F0:A1 A1 A1 A1 8C A1 AB AC <B8>
06F8:80 80 6B 80 80 80 80 80 <5E>
0700:80 80 6C 00 6C A6 8C 8C <99>
0708:A1 8C E2 8C E2 8C 8C 8C <B1>
0710:8C 8C 8C 8C 8C 02 80 8C <88>
0718:8C 8C 8C 8C 8C 8C 8C 8C <C1>
0720:8C A1 6A A1 8C 8C 8C 8C <07>
0728:8C A1 6D 00 6B A1 8C 8C <D5>
0730:A1 8C 8C 8C 8C 8C 8C 8C <1F>
0738:8C 8C 8C 8C 8C 80 02 8C <5F>
0740:8C 8C 8C 8C 8C 8C 8C 8C <9B>
0748:8C 8C 08 8C 8C 8C 8C 8C <22>
0750:8C 8C 08 A1 08 8C 8C 8C <25>
0758:A1 8C 8C 8C 8C 8C 8C 8C <F7>
0760:8C 8C 8C 8C 8C 80 80 02 <A8>
0768:80 80 80 00 B9 A1 A1 A1 <32>
0770:A1 A1 A1 A1 A1 A1 A1 A1 <BF>
0778:A1 A1 A1 A1 A1 A1 A1 A1 <57>
0780:A1 8C 8C 8C 8C 8C 8C 8C <CE>
0788:8C 8C 8C 8C 8C 80 80 80 <1D>
0790:02 80 80 80 B2 A1 A1 A1 <9D>
0798:A1 A1 A1 A1 A1 A1 A1 A1 <37>
07A0:A1 A1 A1 A1 A1 A1 A1 A1 <4F>
07A8:A1 8C 8C 8C 8C 8C 8C 8C <A6>
07B0:8C 8C 8C 8C 8C 8C 8C 8C <29>
07B8:8C E2 80 80 B2 A1 A1 A1 <0F>
07C0:A1 A1 A1 A1 A1 A1 A1 A1 <EF>
07C8:A1 A1 A1 A1 A1 A1 A1 A1 <E7>
07D0:A1 8C 8C 8C 8C 8C 8C 8C <7F>
07D8:8C 8C 8C 8C 8C 8C 8C 8C <01>
07E0:8C 8C 80 80 B9 A1 A1 A1 <BE>
07E8:A1 A1 A1 A1 A1 A1 A1 A1 <47>
07F0:A1 A1 A1 A1 A1 A1 A1 A1 <3F>
07F8:A1 A1 A1 A1 A1 A1 A1 A1 <D7>
0800:A1 A1 A1 A1 A1 A1 A1 A1 <B5>
0808:A8 AD 80 80 A2 A1 A1 A1 <05>
0810:A1 A1 A1 A1 A1 A1 A1 A1 <A5>
0818:A1 A1 A1 A1 A1 A1 A1 A1 <DD>
0820:A1 A1 A1 A1 A1 A1 A1 A1 <96>
0828:A1 A1 A1 A1 A1 A1 A1 A1 <8E>
0830:A1 A7 80 80 80 A2 A1 A1 <CF>
0838:A1 A1 A1 A1 A1 A1 A1 A1 <7E>
0840:A1 A1 A1 A1 A1 A1 A1 A1 <76>
0848:A1 A1 A1 A1 A1 A1 A1 A1 <6E>
0850:A1 A1 A1 A1 A1 A1 A1 A1 <62>
0858:A1 A1 A7 80 80 80 A2 A1 <C0>
0860:A1 A1 A1 A1 A1 A1 A1 A1 <D5>
0868:A1 A1 A1 A1 A1 A1 A1 A1 <CD>
0870:A1 A1 A1 A1 A1 A1 A1 A1 <45>
0878:A1 A1 A1 A1 A1 A1 A1 A1 <3D>
0880:A1 A1 A1 A7 80 80 80 A2 <D1>
0888:A1 A1 A1 A1 A1 A1 A1 A1 <2D>
0890:A1 A1 A1 A1 A1 A1 A1 A1 <25>
0898:A1 A1 A1 A1 A1 A1 A1 A1 <5D>
08A0:A1 A1 A1 A1 A1 A1 A1 A1 <15>
08A8:A1 A1 A1 A1 A7 80 80 80 <57>
08B0:A2 A1 A1 A1 A1 A1 A1 A1 <8D>
08B8:A1 A1 A1 A1 A1 A1 A1 A1 <FD>
08C0:A1 A1 A1 A1 A1 A1 A1 A1 <F5>
08C8:A1 A1 A1 A1 A1 A1 A1 A1 <ED>
08D0:A1 A1 A1 A1 A1 A7 80 80 <9A>
08D8:80 A2 A1 A1 A1 A1 A1 A1 <AD>
08E0:A1 A1 A1 A1 A1 A1 A1 A1 <55>
08E8:A1 A1 A1 A1 A1 A1 A1 A1 <4D>
08F0:A1 A1 A1 A1 A1 A1 A1 A1 <C5>
08F8:A1 A1 A1 A1 A1 A1 A7 80 <A8>
0900:80 80 A2 A1 A1 A1 A1 A1 <24>
0908:A1 A1 A1 A1 A1 A1 A1 A1 <A3>
0910:A1 A1 A1 A1 A1 A1 A1 A1 <AB>
0918:A1 A1 A1 A1 A1 A1 A1 A1 <D3>
0920:A1 A1 A1 A1 A1 A1 A1 A7 <A2>
0928:00 00 00 A2 A1 A1 A1 A1 <C2>
0930:A1 A1 A1 A1 A1 A1 A1 A1 <8C>
0938:A1 A1 A1 A1 A1 A1 A1 A1 <74>
0940:A1 A1 A1 A1 A1 A1 A1 A1 <7C>
0948:A1 A1 A1 A1 A1 A1 A1 A3 <66>
0950:00 00 00 A6 A1 A1 A1 A1 <38>
0958:A1 A1 A1 A1 A1 A1 A1 A1 <94>
0960:A1 A1 A1 A1 A1 A1 A1 A1 <DB>
0968:A1 A1 A1 A1 A1 A1 A1 A1 <C3>
0970:A1 A1 A1 A1 A1 A1 A3 00 <AE>
0978:00 00 A6 A1 A1 A1 A1 A1 <DB>
0980:A1 A1 A1 A1 A1 A1 A1 A1 <3B>
0988:A1 A1 A1 A1 A1 A1 A1 A1 <23>
0990:A1 A1 A1 A1 A1 A1 A1 A1 <2B>
0998:A1 A1 A1 A1 B8 A3 00 00 <32>
09A0:00 A6 A1 A1 A1 A1 A1 A1 <BC>
09A8:A1 A1 A1 A1 A1 A1 A1 A1 <03>
09B0:A1 A1 A1 A1 A1 A1 A1 A1 <0B>
09B8:A1 A1 A1 A1 A1 A1 A1 A1 <F3>
09C0:A1 A1 A1 B8 80 00 00 00 <FE>
09C8:A6 A1 A1 A1 A1 A1 A1 A1 <61>
09D0:A1 A1 A1 A1 A1 A1 A1 A1 <EB>
09D8:A1 A1 A1 A1 A1 A1 A1 A1 <13>
09E0:A1 A1 A1 A1 A1 A1 A1 A1 <5B>
09E8:A1 A1 A9 80 00 00 80 A6 <67>
09F0:A1 A1 A1 A1 A1 A1 A1 A1 <CB>
09F8:A1 A1 A1 A1 A1 A1 A1 A1 <B3>
0A00:A1 A1 A1 A1 A1 A1 A1 A1 <A9>
0A08:A1 A1 A1 A1 A1 A1 06 6E <38>
0A10:6D 6C 6B 6A 6F 6E 6D 6C <65>
0A18:6B 6A 07 A1 A1 A1 A1 A1 <D5>
0A20:A1 A1 A1 A1 A1 A1 A1 A1 <8A>
0A28:A1 A1 A1 A1 A1 A1 A1 A1 <82>
0A30:A1 A1 A1 A1 A1 A1 A1 A1 <9A>
0A38:A1 A3 00 00 00 A6 A1 A1 <C2>
0A40:A1 A1 A1 A1 A1 A1 A1 A1 <6A>
0A48:A1 A1 A1 A1 A1 A1 A1 A1 <66>
0A50:A1 A1 A1 A1 A1 A1 A1 A1 <7A>
0A58:A1 A1 A1 A1 A1 A1 A1 A1 <92>
0A60:A3 00 00 80 A6 A1 A1 A1 <5F>
0A68:A1 A1 A1 A1 A1 A1 A1 A1 <C1>
0A70:A1 A1 A1 A1 A1 A1 A1 A1 <59>
0A78:A1 A1 A1 A1 A1 A1 A1 A1 <31>
0A80:A1 A1 A1 A1 A1 A1 A1 A3 <2B>
0A88:00 00 00 A6 A1 A1 A1 A1 <62>
0A90:A1 A1 A1 A1 A1 A1 A1 A1 <39>
0A98:A1 A1 A1 A1 A1 A1 A1 A1 <51>
0AA0:A1 A1 A1 A1 A1 A1 A1 A1 <09>
0AA8:A1 A1 A1 A1 A1 A1 A3 00 <64>
0AB0:00 00 A6 A1 A1 A1 A1 A1 <11>
0AB8:A1 A1 A1 A1 A1 A1 A1 A1 <F1>
0AC0:A1 A1 A1 A1 A1 A1 A1 A1 <E9>
0AC8:A1 A1 A1 A1 A1 A1 A1 A1 <E1>
0AD0:A1 A1 A1 A1 A1 A3 80 80 <9E>
0AD8:00 A6 A1 A1 A1 A1 A1 A1 <E2>
0AE0:A1 A1 A1 A1 A1 A1 A1 A1 <49>
0AE8:A1 A1 A1 A1 A1 A1 A1 A1 <41>
0AF0:F0 79 17 74 00 00 00 00 <B4>
0AF8:00 00 00 00 00 00 00 00 <0C>
0B00:00 00 00 00 00 00 00 00 <16>
```

»Level 5« (Schluß)

Die Macht des Geistes

Messerscharfe Logik und ein gutes Vorstellungsvermögen sind die Voraussetzung, um unser Denkspiel für zwei zu meistern.

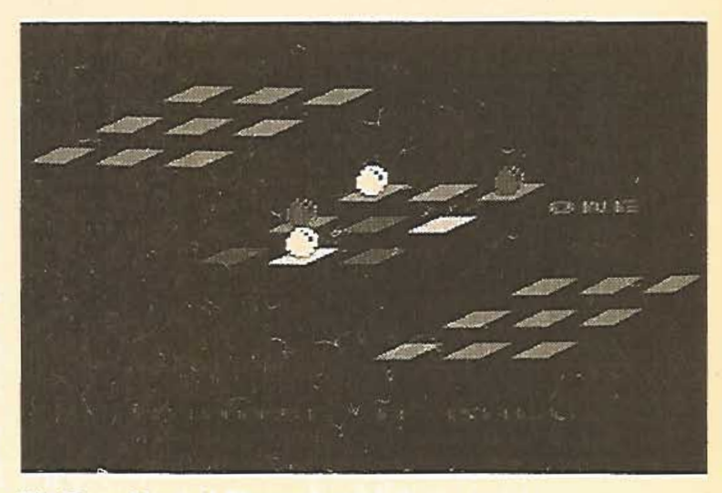
Es ist später Abend. Die Straßen sind leer, nur in einem einzelnen Haus am Stadtrand sitzen noch zwei Menschen in einem Raum, der in schummriges Licht getaucht ist. Diese beiden Atarianer haben sich intensiv mit unserem Denk- und Strategiespiel »Stratfield« beschäftigt.

Es geht darum, drei Felder in einer Reihe mit Hilfe beweglicher Kugeln in »seiner« Farbe einzufärben. Dabei ist man aber nicht auf eine Ebene beschränkt. Die Felder können auch untereinander auf verschiedenen Ebenen liegen. Diagonale Reihen sind nicht erlaubt.

Sie können Stratfield in zwei Varianten spielen, die Sie vor Spielbeginn einstellen. Bei der ersten Art hat jeder Spieler zwei Kugeln: Spieler 1 die roten, Spieler 2 die weißen. Sie müssen sich genau merken, welche Kugel Nummer 1 und welche Nummer 2 ist, da es während des Spiels nicht angezeigt wird, und Sie vor jedem Zug angeben, welche Kugel Sie bewegen wollen. In der Ausgangsstellung ist die vordere Kugel die Kugel 1. Wenn Sie Variante 2 gewählt haben, wird das Spiel noch vertrackter. Hier hat nicht jeder Spieler eine zweite Kugel, es gibt statt dessen eine Zusatzkugel, die beide Spieler steuern können. Die Zusatzkugel hat für beide Spieler die Nummer 2.

Gezogen wird abwechselnd, die Züge können Sie nicht zurücknehmen. Die Ebenen wechseln Sie durch die Tasten <P> (hinauf) und <;> (hinunter). Auf der Ebene bewegen Sie die Kugel mit den Cursor-Tasten (ohne <CONTROL>).

Wie wird Stratfield gespielt? Wenn Sie mit einer Kugel zum ersten Mal auf ein Feld kommen, wird es für Sie reserviert. Erst wenn Sie es zum zweiten Mal besetzen, wird es in Ihrer Farbe eingefärbt. Der Gegner kann die Felder aber auf dem gleichen Weg wieder umfärben.


Ein Kampf in drei Dimensionen

Das Programm ist in Turbo Basic geschrieben. Zum Spielen empfiehlt sich ein Farbmonitor.

(Thomas Tennert/gn/hf)

Steckbrief	
Programm:	Stratfield
Sprache:	Turbo-Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Diskette, Kassette

```
10 REM *** STRATFIELD ***               <HW>
20 REM *** WRITTEN BY ***               <ZX>
30 REM *** TH.TENNERT ***               <KZ>
40 REM ***  (c) 1987  ***               <CF>
50 --                                   <ZA>
60 CLR :POKE 559,0                      <JQ>
70 EXEC DECLARE                         <YV>
80 IF PEEK(40960)<>94 THEN EXEC CHSET   <ZS>
90 EXEC TITEL:POKE 559,62               <EI>
100 EXEC SOUND                          <UC>
110 EXEC DIF                            <HI>
120 IF O=2                              <PR>
130 CO=8:IO=2:PO=1280:PKO=53249:PMO=0:EX
EC STARTKO_2                            <BV>
140 ELSE                                <XV>
150 CO=36:IO=4:PO=1792:PKO=53251:EXEC ST
ARTKO                                   <JO>
160 ENDIF                               <VC>
170 EXEC GRAFIK                         <YA>
180 POKE 53277,3:POKE 623,1:POKE 731,255 <GB>
190 POKE 712,0:POKE 710,4:POKE 709,180:P
OKE 708,148                             <GL>
200 EXEC INIT                           <CS>
210 EXEC PMBAU                          <MC>
220 REPEAT                              <DB>
230 EXEC SPWAHL                         <JF>
240 YEA(I)=YE:XA(I)=X:YA(I)=Y:XS(I)=XF:Y
S(I)=YF:XCA(I)=XC:YCA(I)=YC             <NC>
250 EXEC GEWINNER                       <AO>
260 UNTIL G=1                           <JH>
270 EXEC WERTUNG                        <NL>
280 EXEC SOUND                          <UT>
290 RUN                                 <FA>
300 --                                  <JE>
310 PROC TAST                           <JP>
320 YE=YEA(I):X=XA(I):Y=YA(I):XF=XS(I):Y
F=YS(I):YC=YCA(I):XC=XCA(I):PLP=0       <OB>
330 REPEAT                              <DE>
340 GET T                               <DO>
350 UNTIL T=80 OR T=59 OR T=45 OR T=61 O
R T=43 OR T=42                          <PB>
360 FOR Z=1 TO 80 STEP 2:SOUND 0,50+Z,10
,8:NEXT Z:SOUND 0,0,0,0                 <SB>
370 IF T=80 AND YE>1 THEN YE=YE-1:Y=Y-48
:YC=YC-6:XC=XC-5:X=X-40:PLP=1           <LT>
380 IF T=59 AND YE<3 THEN YE=YE+1:Y=Y+48
:YC=YC+6:XC=XC+5:X=X+40:PLP=1           <WQ>
390 IF T=45 AND YF>1 THEN YF=YF-1:Y=Y-16
:X=X+16:YC=YC-2:XC=XC+2:PLP=1           <DG>
400 IF T=61 AND YF<3 THEN YF=YF+1:Y=Y+16
:X=X-16:YC=YC+2:XC=XC-2:PLP=1           <AL>
410 IF T=43 AND XF>1 THEN XF=XF-1:X=X-16
:XC=XC-2:PLP=1                          <AI>
420 IF T=42 AND XF<3 THEN XF=XF+1:X=X+16
:XC=XC+2:PLP=1                          <RB>
430 IF PLP=1 THEN EXEC PLPOSPRUEF       <LW>
440 ENDPROC                             <KO>
450 --                                  <JP>
460 PROC TAST_2                         <MA>
470 IF T=80 THEN YE=YE+1:Y=Y+48:YC=YC+6:
XC=XC+5:X=X+40                          <ES>
480 IF T=59 THEN YE=YE-1:Y=Y-48:YC=YC-6:
XC=XC-5:X=X-40                          <VX>
490 IF T=45 THEN YF=YF+1:Y=Y+16:X=X-16:Y
C=YC+2:XC=XC-2                          <NE>
500 IF T=61 THEN YF=YF-1:Y=Y-16:X=X+16:Y
C=YC-2:XC=XC+2                          <MF>
510 IF T=43 THEN XF=XF+1:X=X+16:XC=XC+2  <BC>
```

Ein fesselndes Denkspiel für zwei Spieler: Stratfield

```
520 IF T=42 THEN XF=XF-1:X=X-16:XC=XC-2    <HR>
530 ENDPROC                                <KN>
540 --                                     <JO>
550 PROC SPWAHL                            <MX>
560 FOR Z=-10 TO 10 STEP 2:SOUND 0,ABS(Z
),12,8:PAUSE 5:NEXT Z:SOUND 0,0,0,0        <LQ>
570 IF SP=0                                <WO>
580 SP=1:POSITION 15,7:? #6;"ONE":POSITI
ON 1,9:? #6;"▴▴▴"                          <JR>
590 C=1:C1=90:C2=89                        <YA>
600 EXEC SP1                               <GM>
610 ELSE                                   <XU>
620 SP=0:POSITION 1,9:? #6;"two":POSITIO
N 15,7:? #6;"▴▴▴"                          <IJ>
630 C=2:C1=122:C2=121                      <MP>
640 EXEC SP2                               <HG>
650 ENDIF                                  <VF>
660 ENDPROC                                <KU>
670 --                                     <JV>
680 PROC SP1                               <KN>
690 REPEAT                                 <DT>
700 POSITION 0,20:? #6;"▴▴▴CHOOSE▴A▴BALL
▴▴▴▴"                                      <JN>
710 GET T                                  <DM>
720 UNTIL T=49 OR T=50                     <RL>
730 POSITION 0,20:? #6;"NUMBER▴";T-48;"▴
WAS▴CHOSEN"                                <ST>
740 IF T=49                                <QM>
750 I=1:EXEC TAST                          <IK>
760 IF FR=1                                <TR>
770 MOVE ADR(LE$),PMB+1024,256:MOVE ADR(
LO$),PMB+1024+Y,14:POKE 53248,X:EXEC E12
3                                          <TK>
780 ELSE                                   <YJ>
790 IF PLP=1 THEN EXEC TAST_2              <NM>
800 ENDIF                                  <UX>
810 ELSE                                   <XW>
820 I=2:EXEC TAST                          <IM>
830 IF FR=1                                <TM>
840 MOVE ADR(LE$),PMB+1280,256:MOVE ADR(
LO$),PMB+1280+Y,14:POKE 53249,X:EXEC E12
3                                          <HG>
850 ELSE                                   <YE>
860 IF PLP=1 THEN EXEC TAST_2              <NH>
870 ENDIF                                  <VL>
880 ENDIF                                  <VN>
890 ENDPROC                                <LC>
900 --                                     <JK>
910 PROC SP2                               <KO>
920 REPEAT                                 <DI>
930 POSITION 0,20:? #6;"▴▴▴CHOOSE▴A▴BALL
▴▴▴▴"                                      <JV>
940 GET T                                  <DU>
950 UNTIL T=49 OR T=50                     <RT>
960 POSITION 0,20:? #6;"NUMBER▴";T-48;"▴
WAS▴CHOSEN"                                <TB>
970 IF T=49                                <QU>
980 I=3:EXEC TAST                          <JG>
990 IF FR=1                                <TZ>
1000 MOVE ADR(LE$),PMB+1535,256:MOVE ADR
(LO$),PMB+1535+Y,14:POKE 53250,X:EXEC E1
23                                         <XZ>
1010 ELSE                                  <SX>
1020 IF PLP=1 THEN EXEC TAST_2             <HU>
1030 ENDIF                                 <SP>
1040 ELSE                                  <TG>
1050 I=IO:EXEC TAST                        <IN>
1060 IF FR=1                               <TC>
1070 MOVE ADR(LE$),PMB+PO,256:MOVE ADR(L
O$),PMB+PO+Y,14:POKE PKO,X:EXEC E123       <DE>
1080 ELSE                                  <TS>
1090 IF PLP=1 THEN EXEC TAST_2             <IP>
1100 ENDIF                                 <SI>
1110 ENDIF                                 <SL>
1120 ENDPROC                               <OK>
1130 --                                    <VP>
1140 PROC PLPOSPRUEF                       <SW>
1150 FR=0                                  <UY>
1160 IF T<>80 OR T<>59                     <CD>
1170 IF YE=1 AND PE1(XF,YF)=0 THEN FR=1:
PE1(XF,YF)=1:PE1(XS(I),YS(I))=0            <OF>
1180 IF YE=2 AND PE2(XF,YF)=0 THEN FR=1:
PE2(XF,YF)=1:PE2(XS(I),YS(I))=0            <TJ>
1190 IF YE=3 AND PE3(XF,YF)=0 THEN FR=1:
PE3(XF,YF)=1:PE3(XS(I),YS(I))=0            <YH>
1200 ENDIF                                 <SK>
1210 IF T=80 OR T=59                       <GD>
1220 IF YE=1 AND PE1(XF,YF)=0 THEN FR=1:
PE1(XF,YF)=1:PE2(XF,YF)=0                  <VP>
1230 IF YE=2 AND YEA(I)=1 AND PE2(XF,YF)
=0 THEN FR=1:PE2(XF,YF)=1:PE1(XF,YF)=0     <CA>
1240 IF YE=2 AND YEA(I)=3 AND PE2(XF,YF)
=0 THEN FR=1:PE2(XF,YF)=1:PE3(XF,YF)=0     <JJ>
1250 IF YE=3 AND PE3(XF,YF)=0 THEN FR=1:
PE3(XF,YF)=1:PE2(XF,YF)=0                  <BS>
1260 ENDIF                                 <TC>
1270 ENDPROC                               <PB>
1280 --                                    <WG>
1290 PROC E123                             <WM>
1300 IF YE=1:IF E1(XF,YF)<>C:E1(XF,YF)=C
:COLOR C3:PLOT XC,YC:COLOR C4:PLOT XC+1,
YC:SE1(XF,YF)=0                            <GL>
1310 ELSE :COLOR C1:PLOT XC,YC:COLOR C2:
PLOT XC+1,YC:SE1(XF,YF)=C:M=1              <XP>
1320 ENDIF :ENDIF                          <OB>
1330 IF YE=2:IF E2(XF,YF)<>C:E2(XF,YF)=C
:COLOR C3:PLOT XC,YC:COLOR C4:PLOT XC+1,
YC:SE2(XF,YF)=0                            <ML>
1340 ELSE :COLOR C1:PLOT XC,YC:COLOR C2:
PLOT XC+1,YC:SE2(XF,YF)=C:M=1              <AC>
1350 ENDIF :ENDIF                          <OK>
1360 IF YE=3:IF E3(XF,YF)<>C:E3(XF,YF)=C
:COLOR C3:PLOT XC,YC:COLOR C4:PLOT XC+1,
YC:SE3(XF,YF)=0                            <SL>
1370 ELSE :COLOR C1:PLOT XC,YC:COLOR C2:
PLOT XC+1,YC:SE3(XF,YF)=C:M=1              <CP>
1380 ENDIF :ENDIF                          <OT>
1390 IF M=1 THEN FOR Z=1 TO 15:SOUND 0,6
0,14,10:PAUSE 3:SOUND 0,0,0,0:NEXT Z:M=0   <SR>
1400 FR=0                                  <UP>
1410 ENDPROC                               <ON>
1420 --                                    <VS>
1430 PROC GRAFIK                           <OA>
1440 GRAPHICS 17:POKE 756,PAGE             <AW>
1450 ? #6;"▴▴▴zyzyzy▴▴▴▴▴▴▴▴▴▴▴"           <CV>
1460 ? #6;"▴▴zyzyzy▴▴▴▴▴▴▴▴▴▴▴▴"           <GO>
1470 ? #6;"zyzyzy▴▴▴▴▴▴▴▴▴▴▴▴▴▴"           <KH>
1480 ? #6;"▴▴▴▴▴▴▴▴▴zyzyzy▴▴▴▴▴"           <UD>
1490 ? #6;"▴▴▴▴▴▴▴zyzyzy▴▴▴▴▴▴▴"           <XW>
1500 ? #6;"▴▴▴▴▴zyzyzy▴▴▴▴▴▴▴▴▴"           <AN>
1510 ? #6;"▴▴▴▴▴▴▴▴▴▴▴▴▴▴zyzyzy"           <KJ>
1520 ? #6;"▴▴▴▴▴▴▴▴▴▴▴▴zyzyzy▴▴"           <OC>
1530 ? #6;"▴▴▴▴▴▴▴▴▴▴zyzyzy▴▴▴▴"           <RV>
1540 ENDPROC                               <OY>
1550 --                                    <WD>
1560 PROC DECLARE                          <EF>
1570 DIM YEA(4),XS(5),YS(5),XA(5),YA(5),
XCA(5),YCA(5),E1(4,4),E2(4,4),E3(4,4),G$
(5)                                        <UV>
1580 DIM LE$(256),LO$(18),PE1(4,4),PE2(4
,4),PE3(4,4),SE1(4,4),SE2(4,4),SE3(4,4):
C3=218:C4=217                              <WC>
1590 RESTORE 1630                          <NL>
1600 FOR Z=1 TO 14                         <ML>
1610 READ W:LO$(Z,Z)=CHR$(W)               <NE>
1620 NEXT Z                                <MM>
1630 DATA▴28,22,58,59,61,125,125,127,127
,126,126,62,60,28                          <RM>
1640 ENDPROC                               <PA>
1650 --                                    <WF>
1660 PROC INIT                             <OU>
1670 PB=PEEK(106)-32:POKE 54279,PB         <GF>
1680 PMB=PB*256:POKE 559,62                <GL>
1690 FOR Z=53256 TO 53259:POKE Z,0:NEXT
Z                                          <DJ>
1700 FOR Z=1024 TO 1792 STEP 256:MOVE AD
R(LE$),PMB+Z,256:NEXT Z                    <KG>
1710 POKE 704,36:POKE 705,CO:POKE 706,14
:POKE 707,14                               <NA>
1720 ENDPROC                               <OW>
1730 --                                    <WB>
1740 PROC STARTKO                          <GV>
```

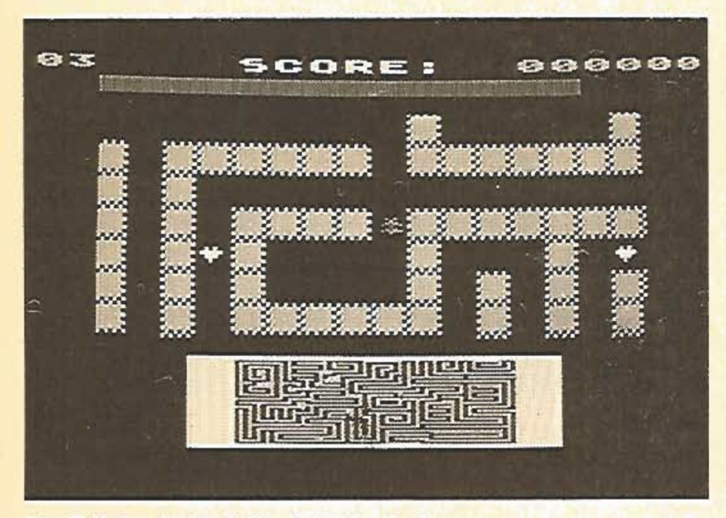
```
1750 XA(1)=91:XA(2)=155:XA(3)=123:XA(4)=
123:YA(1)=103:YA(2)=71:YA(3)=103:YA(4)=7
1                                          <BC>
1760 XS(1)=1:XS(2)=3:XS(3)=3:XS(4)=1:YS(
1)=3:YS(2)=1:YS(3)=3:YS(4)=1               <VB>
1770 FOR Z=1 TO 4:YEA(Z)=2:NEXT Z:SP=0     <ZV>
1780 PE2(1,3)=1:PE2(3,3)=1:PE2(3,1)=1:PE
2(1,1)=1                                   <DX>
1790 XCA(1)=5:XCA(2)=13:XCA(3)=9:XCA(4)=
9:YCA(1)=10:YCA(2)=6:YCA(3)=10:YCA(4)=6    <WK>
1800 ENDPROC                               <OS>
1810 --                                    <VX>
1820 PROC STARTKO_2                        <EZ>
1830 XA(1)=91:XA(2)=123:XA(3)=155:YA(1)=
103:YA(2)=87:YA(3)=71                      <NQ>
1840 XS(1)=1:XS(2)=2:XS(3)=3:YS(1)=3:YS(
2)=2:YS(3)=1                               <ON>
1850 FOR Z=1 TO 4:YEA(Z)=2:NEXT Z:SP=0     <ZR>
1860 PE2(1,3)=1:PE2(2,2)=1:PE2(3,1)=1      <WC>
1870 XCA(1)=5:XCA(2)=9:XCA(3)=13:YCA(1)=
10:YCA(2)=8:YCA(3)=6                       <VV>
1880 ENDPROC                               <PQ>
1890 --                                    <WV>
1900 PROC PMBAU                            <AD>
1910 MOVE ADR(LO$),PMB+1024+103,14:MOVE
ADR(LO$),PMB+1280+YA(2),14:MOVE ADR(LO$)
,PMB+1535+YA(3),14                         <IK>
1920 MOVE ADR(LO$),PMB+1792+YA(4),14:POK
E 53248,91:POKE 53249,XA(2):POKE 53250,X
A(3):POKE 53251,XA(4)                      <HP>
1930 ENDPROC                               <PD>
1940 --                                    <WI>
1950 PROC GEWINNER                         <XW>
1960 FOR Z=1 TO 3                          <ER>
1970 IF SE1(Z,1)=C AND SE1(Z,2)=C AND SE
1(Z,3)=C THEN G=1:EXIT                     <OG>
1980 IF SE2(Z,1)=C AND SE2(Z,2)=C AND SE
2(Z,3)=C THEN G=1:EXIT                     <RJ>
1990 IF SE3(Z,1)=C AND SE3(Z,2)=C AND SE
3(Z,3)=C THEN G=1:EXIT                     <UM>
2000 IF SE1(1,Z)=C AND SE1(2,Z)=C AND SE
1(3,Z)=C THEN G=1:EXIT                     <WA>
2010 IF SE2(1,Z)=C AND SE2(2,Z)=C AND SE
2(3,Z)=C THEN G=1:EXIT                     <ZD>
2020 IF SE3(1,Z)=C AND SE3(2,Z)=C AND SE
3(3,Z)=C THEN G=1:EXIT                     <CG>
2030 NEXT Z                                <ME>
2040 IF G=0                                <JB>
2050 FOR Z=1 TO 3:FOR ZZ=1 TO 3            <XE>
2060 IF SE1(Z,ZZ)=C AND SE2(Z,ZZ)=C AND
SE3(Z,ZZ)=C THEN G=1:EXIT                  <TR>
2070 NEXT ZZ:NEXT Z                        <XZ>
2080 ENDIF                                 <TF>
2090 ENDPROC                               <PE>
2100 --                                    <VH>
2110 PROC WERTUNG                          <HP>
2120 FOR Z=1 TO 4:POKE 53247+Z,0:NEXT Z    <TD>
2130 IF C=1:G$(1,5)="▲red▲":ELSE :G$(1,5
)="white":ENDIF                            <HG>
2140 POSITION 7,19:? #6;G$(1,5)            <UF>
2150 POSITION 0,20:? #6;"▲▲▲▲▲▲has▲won▲▲   <TO>
2160 POSITION 4,22:? #6;"▲▲▲▲▲▲PRESS▲STA
RT"                                        <WI>
2170 ENDPROC                               <PA>
2180 --                                    <WF>
2190 PROC TITEL                            <GK>
2200 GRAPHICS 7+16:POKE 559,0:POKE 756,P
AGE                                        <BQ>
2210 COLOR 1:CIRCLE 40,50,20,40:COLOR 2:
CIRCLE 45,30,5,7                           <AU>
2220 COLOR 1:PAINT 40,50:COLOR 2:PAINT 4
5,30                                       <EJ>
2230 COLOR 3:PLOT 57,70:DRAWTO 150,70:DR
AWTO 100,30:DRAWTO 57,30                   <ZQ>
2240 PLOT 90,70:DRAWTO 70,30:PLOT 119,45
:DRAWTO 59,45                              <BJ>
2250 PAINT 95,35:PAINT 60,60               <SJ>
2260 COLOR 2:CIRCLE 130,20,8,12            <DK>
2270 COLOR 1:CIRCLE 128,16,1,1             <ZE>
2280 COLOR 2:PAINT 130,20                  <RE>
2290 FOR Z=0 TO 8 STEP 2:PLOT 0,Z:DRAWTO
159-Z*7,Z:NEXT Z                           <IE>
2300 TEXT 0,0,"STRATFIELD"                 <AH>
2310 TEXT 60,71,"VON":TEXT 49,81,"THOMAS
▲TENNERT"                                  <AS>
2320 COLOR 3:TEXT 40,89,"PRESS▲START"      <AX>
2330 ENDPROC                               <OS>
2340 --                                    <VX>
2350 PROC DIF                              <PI>
2360 CLS #6:GRAPHICS 1+16:O=1:POKE 756,P
AGE                                        <DW>
2370 REPEAT                                <FB>
2380 POSITION 4,5:? #6;"press▲select▲▲▲▲
▲for▲choosing▲game▲▲":POSITION 9,8:? #6;
O                                          <SU>
2390 POSITION 4,13:? #6;"PRESS▲OPTION"     <TH>
2400 PK=PEEK(53279)                        <YP>
2410 IF PK=5 THEN O=O+1:FOR Z=-80 TO 80
STEP 8:SOUND 0,ABS(60+Z),10,10:PAUSE 3:N
EXT Z:IF O=3 THEN O=1                      <TP>
2420 SOUND 0,0,0,0                         <HL>
2430 UNTIL PK=3                            <KO>
2440 ENDPROC                               <OX>
2450 --                                    <WC>
2460 PROC SOUND                            <JJ>
2470 DSOUND                                <JD>
2480 TRAP 2480:RESTORE 2560                <EY>
2490 READ H,L:P=P+1                        <YZ>
2500 SOUND 0,H,10,L:SOUND 3,255,12,15:SO
UND 3,0,0,0                                <CU>
2510 IF P>24 THEN SOUND 2,60+L,12,2        <QE>
2520 IF P=94 THEN P=0                      <MR>
2530 PAUSE 9:SOUND 2,0,0,0                 <JW>
2540 IF PEEK(53279)=6 THEN DSOUND :P=0:E
NDPROC                                     <VR>
2550 GOTO 2490                             <UI>
2560 DATA▲108,7,96,8,91,9,81,10,0,0,91,8
,96,9,0,0,108,7,96,8,91,9,81,10,0,0,91,8
,96,9,0,0,81,9,91,8,96,7,81,9              <FG>
2570 DATA▲0,0,81,9,0,0,72,9,0,0,0,0,0,0,
81,10,91,9,81,10,0,0,96,8,0,0,91,10,0,0,
0,0                                        <WE>
2580 DATA▲96,10,91,9,108,11,0,0,96,11,91
,12,0,0,81,10,96,10,0,0,0,0,               <SH>
2590 ENDPROC                               <PO>
2600 --                                    <VR>
2610 PROC CHSET                            <CI>
2620 RESTORE 2650                          <ON>
2630 PAGE=PEEK(106)-32:CHS=PAGE*256        <UB>
2640 FOR Z=0 TO 26:READ C:FOR Z1=0 TO 7:
READ A:POKE CHS+Z1+C*8,A:NEXT Z1:NEXT Z    <IM>
2650 DATA▲33,0,94,66,98,110,98,98,0        <HU>
2660 DATA▲34,0,60,76,76,98,98,110,0        <WS>
2670 DATA▲35,0,94,78,96,96,102,126,0       <LC>
2680 DATA▲36,0,92,66,98,98,110,108,0       <KT>
2690 DATA▲37,0,94,64,96,110,96,110,0       <AB>
2700 DATA▲38,0,94,70,96,108,96,96,0        <GK>
2710 DATA▲39,0,94,64,96,110,102,126,0      <MD>
2720 DATA▲40,0,66,66,110,98,98,98,0        <IG>
2730 DATA▲41,0,16,16,24,24,24,24,0         <HQ>
2740 DATA▲43,0,70,72,112,108,98,98,0       <NR>
2750 DATA▲44,0,64,64,96,96,98,110,0        <XD>
2760 DATA▲45,0,66,86,106,106,98,98,0       <WP>
2770 DATA▲46,0,82,90,106,106,110,102,0     <QN>
2780 DATA▲47,0,60,70,102,102,98,60,0       <SF>
2790 DATA▲48,0,92,70,102,108,96,96,0       <LR>
2800 DATA▲50,0,92,66,98,108,98,98,0        <MV>
2810 DATA▲51,0,62,70,112,14,98,124,0       <UJ>
2820 DATA▲52,0,126,22,16,24,24,24,0        <VK>
2830 DATA▲53,0,66,66,98,98,110,46,0        <XC>
2840 DATA▲54,0,98,98,66,36,52,8,0          <XZ>
2850 DATA▲55,0,70,70,66,90,60,36,0         <VD>
2860 DATA▲56,0,0,0,0,16,0,0,0              <BJ>
2870 DATA▲57,255,254,252,248,240,224,192
,128                                       <MO>
2880 DATA▲58,1,3,7,15,31,63,127,255        <RL>
2890 DATA▲17,8,24,24,24,24,24,24,24        <HI>
2900 DATA▲18,126,195,196,2,12,48,195,255   <HR>
2910 DATA▲0,0,0,0,0,0,0,0,0                <QH>
2920 ENDPROC                               <PB>
```

Stratfield, ein fesselndes Denkspiel für zwei Spieler (Schluß)

Ein Käfer auf Wanderschaft

Damit der kleine Käfer im großen Labyrinth am Leben bleibt, muß er Nahrung finden und vor den gefährlichen Wächtern fliehen.

Der Käfer Karl lebt in einem großen unübersichtlichen Labyrinth. Er muß die ganze Zeit herumlaufen, um Nahrung zu finden. Diese liegt in Form von kleinen roten Quadraten überall verstreut herum. Es befinden sich aber auch andere Sachen in den Gängen. Herzen bringen Ihnen Punkte ein, wenn Sie den kleinen Käfer steuern.



Ein Teil des Labyrinths bei Wandering

Alles andere ist für Karl lebensgefährlich. Er darf weder die Wände berühren, noch sich von den umherstreifenden Wachen erwischen lassen. Eine Übersichtskarte zeigt an, wo die Wachen gerade sind. Wenn er nicht regelmäßig etwas zu essen findet, verhungert er. Die Balkengrafik zeigt an, wie groß Karls Nahrungsreserve ist. Die Wände passen sich der Zahl von Karls Leben an. Mit seinem dritten Leben kann er sogar durch die Wände hindurchlaufen, wenn er gut zielt. »Wandering« kennt zwei Schwierigkeitsstufen, die man mit der <SELECT>-Taste einstellt. Sie verändern nur die Punktegrenze, ab der Sie ein Extraleben erhalten. Auf Stufe 1 liegt die Grenze bei 1000 Punkten, auf Stufe 2 schon bei 10000 Punkten.

Wenn Sie das Spiel anhalten wollen, drücken Sie die <OPTION>-Taste. Um das Spiel fortzusetzen, genügt ein Druck auf die <SELECT>-Taste. Die <START>-Taste beendet das laufende Spiel.

Wenn Sie das Programm starten, dauert es rund zwei Minuten, bis die Daten gelesen sind. Bitte warten Sie solange, selbst wenn sich auf dem Bildschirm nichts tut.

(Holger Winkler/gn/hf)

Steckbrief	
Programm:	Wandering
Sprache:	eingebautes Basic
Eingabehilfe:	Prüfsummer
Datenträger:	Kassette, Diskette

```
1 REM *** HOLGER WINKLER ***                    <MW>
2 REM *** GERMANENSTR.96 ***                    <JA>
3 REM ***  4690 HERNE 1   ***                   <SM>
10 GRAPHICS 7:POKE 559,0:DIM LAST(1),HIG
H(1):FOR I=1 TO 14:READ WERT,J:POKE WERT
,J:NEXT I:FOR I=26632 TO I+15                   <FH>
20 READ WERT,J:POKE I,WERT:POKE I+80,J:N
EXT I:SOUND 1,0,0,0:FOR I=26880 TO I+773
:READ WERT:POKE I,WERT:NEXT I                   <WT>
30 POKE 623,16:POKE 87,15:POKE 88,0:POKE
 89,144:POKE 705,15                             <SS>
50 FOR I=28674 TO I+79:POKE I,65:POKE I+
80,65:POKE I+160,65:POKE I+3600,65:POKE
I+3680,65:POKE I+3760,65:NEXT I                 <WT>
60 FOR I=28914 TO I+3280 STEP 80:FOR J=0
 TO 7:POKE I+J,65:NEXT J:FOR J=72 TO 79:
POKE I+J,65:NEXT J:NEXT I                       <DX>
80 FOR I=0 TO 7:POKE 35945+I,PEEK(26931+
I):NEXT I:FOR A=1 TO 117:READ WERT,J:FOR
 I=28922+WERT TO I+J:POKE I,2                   <NP>
90 NEXT I:NEXT A:FOR A=1 TO 81:READ WERT
,J:FOR I=28922+WERT TO I+J*80 STEP 80:PO
KE I,2:NEXT I:NEXT A                            <PI>
100 LAST(0)=0:LAST(1)=0:HIGH(0)=0:HIGH(1
)=0:POKE 29614,0                                <RG>
110 FOR I=27904 TO I+471:READ WERT:POKE
I,WERT:NEXT I:FOR I=32582 TO I+15:POKE I
,129:NEXT I:GOSUB 2000:GOSUB 3100               <BW>
120 POKE 1571,3:POKE 560,0:POKE 561,156:
FOR I=0 TO 5:POKE 40511+I,PEEK(32576+I)+
16:POKE 32576+I,16                              <PO>
130 NEXT I:I=4:SETCOLOR 0,I,8:SETCOLOR 1
,I,12:SETCOLOR 2,I,4:SETCOLOR 3,7,4:POKE
 756,152:GOSUB 2910                             <WX>
140 POKE 512,227:POKE 513,159:POKE 54286
,192:POKE 559,62:FOR I=0 TO 7:POKE 53248
+I,0:NEXT I                                     <DY>
150 I=PEEK(53279):IF I=5 THEN 3400              <BP>
160 IF I<>6 AND PEEK(644)=1 THEN 150            <AY>
180 GOSUB 2970:POKE 28222,3-(PEEK(40593)
=34)                                            <GI>
190 POKE 54286,64:POKE 559,0:POKE 512,38
:POKE 513,105:POKE 708,154:POKE 710,56:P
OKE 711,76:POKE 756,224                         <UK>
200 POKE 709,222:POKE 204,0:POKE 205,20:
POKE 206,0:POKE 207,29:POKE 54276,0:POKE
 54277,0:POKE 53248,126                         <KT>
205 POKE 1575,62:I=USR(34868):POKE 36293
,1:POKE 1576,20:POKE 1587,1:POKE 1588,4:
POKE 1589,16:POKE 1590,64                       <RH>
207 FOR I=0 TO 3:POKE 1583+I,0:NEXT I           <GE>
210 RESTORE 1290:FOR I=1536 TO I+32:READ
 WERT:POKE I,WERT:NEXT I:RESTORE 600:FOR
 I=26880 TO I+34:READ WERT                      <KN>
220 POKE I,WERT:POKE I+5702,129:NEXT I:I
=USR(28284):POKE 53278,0:POKE 32568,115:
POKE 32569,99:POKE 32573,90                     <BR>
230 POKE 32570,111:POKE 32571,114:POKE 3
2572,101:POKE 560,0:POKE 561,105:POKE 55
9,62:POKE 54286,192                             <BB>
235 POKE 35761,85:GOSUB 3000:POKE 538,1:
GOSUB 2970                                      <PI>
240 I=PEEK(53279):IF I=6 THEN POKE 1571,
0:GOTO 270                                      <FS>
245 IF I<6 THEN POKE 538,(I=5)                  <IO>
250 POKE 32562,(INT(PEEK(1571)/10))+16:P
OKE 32563,(PEEK(1571)-(PEEK(32562)-16)*1
0)+16                                           <ME>
255 IF PEEK(1571)>PEEK(1582) THEN GOSUB
2970                                            <BA>
260 IF PEEK(1568)=0 THEN 240                    <SG>
270 POKE 538,0:SOUND 0,0,0,0:SOUND 3,0,0
,0:SOUND 2,0,0,0:FOR I=1 TO 3:SOUND 1,20
0,10,10:FOR J=1 TO 300:NEXT J                   <FL>
280 SOUND 1,0,0,0:FOR J=1 TO 250:NEXT J:
NEXT I:IF PEEK(1571)=0 THEN POKE 559,0:G
OTO 120                                         <KF>
290 POKE 559,0:GOTO 200                         <OQ>
490 DATA 710,56,711,74,512,38,513,105,54
286,192,552,18,553,106,708,122,115,0,116
,0,53774,0,53277,3,704,22,54279                 <GC>
```

```
500 DATA 136,255,0,255,0,255,0,255,24,25
5,24,255,0,255,0,255,0,170,0,127,0,254,4
0,127,124,254,56,127,16,254,0,85,0         <NM>
600 DATA 112,112,0,70,50,127,128,16,6,0,
112,119,98,118,119,178,118,119,2,119,119
,82,119,119,162,119,119,242                <DX>
610 DATA 119,119,66,120,87,146,120,129,0
,136                                       <UK>
620 DATA 72,169,104,141,9,212,173,39,6,7
6,177,110,0                                <GU>
630 DATA 10,4,14,14,21,14,14,21,21,14,14
,21,14,14,4,10,0,5,42,31,31,42,5,0,0,40,
21,62,62,21,40,0                           <YU>
640 DATA 10,4,14,30,5,14,14,21,21,14,14,
5,30,14,4,10,0,5,42,31,31,42,9,0,0,36,21
,62,62,21,40,0                             <JX>
650 DATA 10,4,14,30,5,14,15,20,20,15,14,
5,30,14,4,10,0,6,42,31,31,42,9,0,0,36,21
,62,62,21,24,0                             <TH>
660 DATA 10,4,14,31,4,14,15,20,20,15,14,
4,31,14,4,10,0,10,42,31,31,42,10,0,0,36,
21,62,62,21,20,0                           <NJ>
670 DATA 10,4,14,31,4,14,31,4,4,31,14,4,
31,14,4,10,0,10,42,31,31,42,10,0,0,20,21
,62,62,21,20,0                             <EP>
680 DATA 169,12,133,208,169,105,133,209,
162,8,160,0,165,114,208,18,165,203,24,21
6,113,208,145,208,144,8,169,0,200          <XG>
690 DATA 113,208,145,208,136,165,114,240
,18,177,208,216,56,229,203,145,208,176,8
,200,177,208,233,0,145,208,136             <ZK>
700 DATA 200,200,200,202,208,206,96        <HT>
710 DATA 169,80,133,203,133,114,173,120,
2,72,201,14,208,21,165,204,240,4,198,204
,16,13,165,205,240,9,169,15,133            <HC>
720 DATA 204,32,211,105,198,205,169,0,13
3,114,104,72,201,13,208,25,165,204,201,1
5,176,4,230,204,16,15,165,205,201          <EE>
730 DATA 40,176,9,169,0,133,204,32,211,1
05,230,205,169,1,133,203,133,114,104,72,
201,11,208,25,165,206,201,7,176,4          <IT>
740 DATA 230,206,16,15,165,207,201,62,17
6,9,169,0,133,206,32,211,105,230,207,169
,0,133,114,104,201,7,208,21,165            <IT>
750 DATA 206,240,4,198,206,16,13,165,207
,240,9,169,7,133,206,32,211,105,198,207,
165,204,141,5,212,165,206,141,4            <KK>
755 DATA 212,169,0,141,3,210,173,120,2,2
01,15,240,85,160,4,162,0,74,176,1,232,13
6,208,249                                  <MO>
760 DATA 202,208,71,173,120,2,162,255,74
,232,176,252,138,166,115,232,232,232,232
,224,20,144,2,162,0,134,115                <UM>
770 DATA 24,216,101,115,170,240,9,169,0,
24,105,8,202,208,250,170,160,0,189,51,10
5,153,105,140,232,200,192,8,144            <UX>
780 DATA 244,230,116,165,116,74,144,10,1
69,164,141,3,210,169,220,141,2,210         <CU>
790 DATA 162,0,32,0,109,162,1,32,0,109,1
62,2,32,0,109,162,3,32,0,109,238,24,6,17
3,24,6,201,8,144,3,32,73,109               <BV>
800 DATA 173,10,210,133,119,173,10,210,2
01,113,144,249,201,126,176,245,133,120,1
60,0,177,119,208,19,169,139,166            <AS>
810 DATA 20,224,253,176,2,169,204,145,11
9,224,245,176,3,152,145,119                <LQ>
820 DATA 173,4,208,201,4,144,76,169,112,
133,120,169,62,56,216,229,207,24,105,250
,144,2,230,120,166,205                     <AJ>
830 DATA 72,165,204,201,8,144,1,232,104,
232,202,240,10,24,105,80,144,2,230,120,2
02,208,246,166,206,224,3,133,119           <UY>
840 DATA 176,9,165,119,24,105,1,144,2,23
0,120,133,119,169,0,168,177,119,240,238,
201,138,144,4,169,0,145,119                <IP>
850 DATA 165,20,208,26,174,31,6,169,0,15
7,70,127,206,31,6,16,3,238,32,6,169,168,
141,3,210,169,70,141,2,210                 <GW>
860 DATA 173,4,208,201,4,208,5,169,5,141
,33,6,173,33,6,240,31,206,33,6,238,31,6,
169,129,174,31,6,224,16,144,5              <AL>
870 DATA 162,15,142,31,6,157,70,127,141,
2,210,169,166,141,3,210,32,10,110          <TY>
880 DATA 141,30,208,173,199,2,24,105,16,
141,199,2,76,66,136,96                     <QR>
900 DATA 20,0,29,7,81,9,94,6,102,5,191,4
,243,5,253,5,262,3,275,0,353,2,414,13,49
9,2,573,4,583,6,661,2,723,3                <PD>
910 DATA 734,3,741,2,821,2,882,1,885,2,8
91,3,898,1,965,2,974,0,978,5,1043,4,1052
,0,1123,0,1132,12,1203,3                   <CA>
915 DATA 1212,1,1295,0,1297,6,1361,7,137
1,4,1459,8,1520,10,1532,4,1618,8,1935,0    <XC>
920 DATA 515,2,679,24,745,9,841,21,908,8
,1000,10,1014,6,1069,10,1095,5             <RL>
930 DATA 1163,6,1227,14,1254,4,1325,6,13
85,3,1390,5,1397,6,1415,4,1552,3,1557,8,
1573,5,1681,10,1694,1,1847,8               <HG>
940 DATA 1731,9,1774,14,1792,5,1840,3,18
93,9,1939,7,1948,11,2055,8,2093,4,2101,3
,2110,9,2213,9,2255,4,2261,5               <JM>
950 DATA 2268,2,2274,3,2356,1,2371,9,240
5,8,2417,7,2514,5,2522,18,2564,18,525,17
,367,3,374,8,216,4,54,6,140,0              <HF>
960 DATA 2672,9,2683,19,2725,15,3068,2,2
834,26,2889,11,2925,15,3226,6,2995,8,305
1,11,3085,15,3201,4                        <SS>
970 DATA 3156,7,3209,15,3243,19,3314,7,1
487,4,1647,2,1723,6,1885,3,2044,3,2205,3
,2363,4                                    <ZV>
1000 DATA 161,14,323,4,405,2,406,3,328,1
2,170,16,12,9,1612,1,816,3,1377,3          <ZD>
1010 DATA 579,3,180,2,182,0,187,2,109,5,
271,5,433,2,1927,4,2009,4,1931,4           <JT>
1030 DATA 2001,15,1923,14,1765,7,2805,4,
2887,5,2889,3,2013,4,2335,2,1857,2,2019,
2,2820,0                                   <JG>
1040 DATA 2181,0,2662,4,2504,8,825,6,202
6,2,2426,9,907,3,1468,3,2028,2             <WA>
1050 DATA 2428,7,1470,5,2350,8,1632,1,21
92,12,2354,1,2834,5,595,2,37,5,677,3,131
7,0,1637,1,119,8,839,3,1719,2              <ZC>
1060 DATA 2199,3,1161,0,1641,12,1243,1,1
405,1,41,6,123,6,45,5,127,2,128,1,50,3,1
32,4,134,2,142,2                           <ZN>
1070 DATA 1011,3,1811,6,1013,6,1973,2,10
20,8,2460,0,3020,0,942,11,2302,11,1571,1
,1803,6,1809,8                             <PX>
1110 DATA 216,189,12,6,201,0,208,5,173,2
4,6,16,10,201,2,208,23,169,7,56,237,24,6
,24,125                                    <RN>
1120 DATA 8,6,168,169,0,153,24,104,169,2
55,153,56,104,208,32,201,3,189,8,6,170,1
60,8,176,12,30,24,104,62,56,104            <OL>
1130 DATA 232,136,208,246,240,10,94,24,1
04,126,56,104,232,136,208,246,96,32,159,
110,234,234                                <EK>
1140 DATA 162,31,169,0,157,56,104,169,25
5,157,24,104,202,16,243,162,255,232,189,
0,6,133,117,189,4,6,133,118                <SU>
1150 DATA 169,0,160,0,145,117,189,16,6,1
33,117,157,0,6,189,20,6,133,118,157,4,6,
138,24,105,67,160,0,145,117                <LP>
1155 DATA 140,25,6,165,20,197,19,176,5,1
89,12,6,16,15                              <ON>
1160 DATA 238,25,6,172,25,6,189,12,6,89,
26,6,157,12,6,160,80,140,26,6,201,0,240,
17                                         <GF>
1170 DATA 160,0,201,2,240,11,160,1,140,2
6,6,201,3,176,2,160,0,136,16,18,165,117,
56,237,26,6,133,119,165                    <DZ>
1180 DATA 118,133,120,176,2,198,120,16,1
6,165,117,24,109,26,6,133,119,165,118,13
3,120,144,2,230,120,160,0,177,119          <UG>
1190 DATA 201,65,240,170,201,2,240,166,1
65,119,157,16,6,165,120,157,20,6,138,24,
105,71,160,0,145,119,224,3,176,3           <FX>
1200 DATA 76,95,109,96                     <PS>
1210 DATA 173,4,208,201,8,208,3,238,34,6
,173,34,6,240,49,173,68,127,201,17,169,2
1,144,2,169,16,141,68,127                  <BG>
```

»Wandering«, ein Spiel für Leute mit Fingerspitzengefühl

```
1220 DATA 162,3,144,18,254,64,127,189,64
,127,201,26,144,8,169,16,157,64,127,202,
16,238,224,3,176,3,238,35,6              <YO>
1230 DATA 206,34,6,141,36,6,162,0,173,36
,6,240,5,162,168,206,36,6,141,4,210,142,
5,210,169,0,133,77                       <QA>
1240 DATA 173,4,208,10,10,10,10,10,10,24
0,3,238,32,6,173,32,6,240,8,169,50,141,2
6,2,206,35,6,96                          <HP>
1250 DATA 104,169,113,133,120,160,0,132,
119,177,119,201,65,240,8,201,2,240,4,169
,0,145,119,200,208,239                   <XL>
1260 DATA 230,120,165,120,201,127,144,23
1,96                                     <RT>
1270 DATA 165,20,201,9,176,6,165,19,105,
15,133,19,169,0,141,24,6,96,73,3,141,39,
6,141,0,212,104                          <FE>
1280 DATA 64,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0               <HF>
1290 DATA 50,122,103,57,116,125,125,113,
0,8,16,24,0,2,2,0,130,42,23,137,116,125,
125,113,0,0,0,3,2,1,2,15,0               <MW>
2000 REM *** STARTBILD ***               <DZ>
2010 RESTORE 2000                        <HK>
2040 FOR I=39936 TO I+31:READ WERT:POKE
I,WERT:NEXT I                            <AT>
2050 FOR I=38920 TO I+59*8-1:READ WERT:P
OKE I,WERT:NEXT I                        <QJ>
2060 FOR I=39968 TO I+10*40-1:READ WERT:
POKE I,WERT:NEXT I                       <JD>
2070 FOR I=40448 TO I+5*40-1:READ WERT:P
OKE I,WERT:NEXT I                        <EF>
2080 FOR J=0 TO 2:FOR I=40701+J*40 TO I+
12:READ WERT:POKE I,WERT:NEXT I:NEXT J   <QL>
2170 FOR I=40928 TO I+26:READ WERT:POKE
I,WERT:NEXT I:POKE 512,227:POKE 513,159:
POKE 54286,192                           <OT>
2180 RETURN                              <PU>
2190 DATA 112,112,112,68,32,156,4,4,4,4,
4,4,4,4,4,4,132,4,5,5,5,4,4,4,4,4,4,4,4,
65,0,156                                 <QC>
2200 DATA 85,85,85,85,85,87,86,86,85,85,
85,85,85,255,255,85,85,85,85,85,85,213,2
13,213                                   <BZ>
2210 DATA 213,213,213,213,213,213,213,21
3,213,213,85,85,85,85,85,85,85,170,170,8
5,85,85,85,85                            <MX>
2220 DATA 86,86,86,85,85,85,85,85,86,86,
86,86,86,86,86,86                        <BT>
2230 DATA 85,85,85,85,85,85,85,85        <ZB>
2240 DATA 217,217,217,217,217,217,217,21
7,217,217,217,218,213,213,213,255,217,21
7,217,217,89,89,89,249                   <HG>
2250 DATA 217,217,217,218,85,85,85,255   <UW>
2260 DATA 170,213,213,213,223,217,217,21
7,170,85,85,85,255,85,85,85,169,89,89,89
,217,217,217,217                         <HW>
2270 DATA 218,213,213,213,215,217,217,21
7,217,89,89,89,217,217,217,217           <LC>
2280 DATA 169,217,217,217,217,217,217,21
7,217,217,217,217,217,217,217,245        <AT>
2290 DATA 85,149,101,101,217,217,217,217
,85,85,85,170,85,85,85,255,217,217,217,8
9,89,101,101,149                         <IE>
2300 DATA 169,89,89,89,245,85,85,85,85,8
5,85,169,89,89,89,245                    <DJ>
2310 DATA 218,213,213,213,214,213,213,21
7,170,85,85,85,255,149,101,89,86,213,117
,93,87,85,85,85                          <QW>
2320 DATA 85,149,101,89,89,217,121,89,21
7,89,89,89,249,85,85,85,106,213,213,213,
255,85,85,85                             <SP>
2330 DATA 85,127,119,119,119,119,119,127
,85,93,125,93,93,93,93,93,85,127,87,87,1
27,117,117,127                           <VW>
2340 DATA 85,127,87,87,127,87,87,127,85,
119,119,119,127,87,87,87,85,127,117,117,
127,87,87,127                            <HK>
2350 DATA 85,127,117,117,127,119,119,127
,85,127,87,87,93,93,117,117,85,127,119,1
19,127,119,119,127                       <CZ>
2360 DATA 85,127,119,119,127,87,87,127   <VN>
2370 DATA 85,101,101,101,101,101,101,105
,85,85,85,169,153,153,153,170,85,85,85,1
06,101,106,86,106                        <CU>
2380 DATA 85,89,89,106,89,89,89,89,85,85
,85,106,101,101,101,106,85,85,85,106,102
,102,102,106                             <OW>
2390 DATA 85,85,85,89,90,89,89,89,85,85,
85,106,102,106,101,106,85,85,85,85,90,85
,90,85                                   <NB>
2400 DATA 85,101,101,106,102,102,102,102
,85,85,85,89,85,89,89,89,85,85,85,106,10
2,106,86,90                              <NN>
2410 DATA 85,85,127,119,119,127,117,117,
85,85,117,125,117,117,85,85,85,127,119,1
27,117,127,85,85                         <XX>
2420 DATA 85,127,117,127,87,127,85,85,11
7,117,253,117,117,117,85,85,85,85,253,22
1,221,255,85,85                          <LP>
2500 DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,3                                <OC>
2510 DATA 8,1,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,3,4                                <XL>
2520 DATA 8,8,9,9,9,9,9,9,9,9,9,9,9,9,
9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,
9,9,9,4,4                                <VA>
2530 DATA 8,8,9,9,9,9,19,9,19,14,15,16
,14,15,16,14,15,21,14,15,24,14,15,16,19,
14,15,16,14,15,16,9,9,9,9,9,9,4,4        <YV>
2540 DATA 8,8,9,9,9,9,10,9,10,10,9,10,
10,9,10,10,9,10,10,9,9,10,9,10,10,10,9,1
0,10,9,9,9,9,9,9,9,4,4                   <OO>
2550 DATA 8,8,9,9,9,9,10,19,10,17,15,1
8,10,9,10,10,9,10,17,15,24,26,27,30,10,1
0,9,10,10,31,16,9,9,9,9,9,9,4,4          <KN>
2560 DATA 8,8,9,9,9,9,11,13,12,20,9,20
,20,9,20,11,22,23,11,22,25,20,28,29,20,2
0,9,20,11,22,12,9,9,9,9,9,9,4,4          <OG>
2570 DATA 8,8,9,9,9,9,9,9,9,9,9,9,9,9,
9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,
9,9,9,4,4                                <VP>
2580 DATA 8,7,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,5,4                                <YW>
2590 DATA 7,6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,6,5                                <EV>
2600 DATA 0,0,0,0,0,0,0,1,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,3,0,0,0,0,
0,0,0,0,0                                <XS>
2610 DATA 0,0,0,0,0,0,0,8,9,42,43,44,4
5,9,44,46,47,48,49,9,50,9,32,32,32,32,32
,32,9,4,0,0,0,0,0,0,0,0                  <OS>
2620 DATA 0,0,0,0,0,0,0,8,9,51,52,53,5
1,9,44,46,47,48,49,9,50,9,32,32,32,32,32
,32,9,4,0,0,0,0,0,0,0,0                  <JY>
2630 DATA 0,0,0,0,0,0,0,8,9,9,9,9,9,44
,49,42,49,46,45,9,50,9,9,9,33,9,9,9,9,4,
0,0,0,0,0,0,0,0,0                        <BI>
2635 DATA 0,0,0,0,0,0,0,7,6,6,6,6,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,6,5,0,0,0,0,
0,0,0,0,0                                <MV>
2640 DATA 1,2,2,2,2,2,2,2,2,2,2,2,3      <SM>
2650 DATA 8,182,183,184,185,185,9,185,18
6,187,183,186,4                          <LY>
2660 DATA 7,6,6,6,6,6,6,6,6,6,6,5        <KW>
2900 DATA 168,172,162,72,173,224,159,141
,10,212,141,22,208,173,225,159,141,23,20
8,173,226,159,141,24,208,104,64          <XP>
2910 A=5:J=0:WERT=0:FOR I=0 TO 5:J=J+(PE
EK(40511+I)-32)*10^A:WERT=WERT+(PEEK(405
51+I)-32)*10^A:A=A-1:NEXT I              <AY>
2920 LAST(PEEK(40593)-33)=J:IF J<=HIGH(P
EEK(40593)-33) THEN RETURN               <RU>
2930 FOR I=0 TO 5:POKE 40551+I,PEEK(4051
1+I):NEXT I:FOR I=0 TO 7:POKE 53248+I,0:
NEXT I                                   <IQ>
2935 HIGH(PEEK(40593)-33)=J:RETURN       <PR>
2940 DATA 0,0,170,127,254,127,170,0      <FC>
```

```
2950 DATA▲0,84,62,124,62,124,42,0                   <CQ>
2960 DATA▲170,127,254,127,254,127,254,85 <FT>
2970 IF PEEK(1571)>3 THEN POKE 1582,PEEK
(1571):RETURN                                       <JJ>
2980 RESTORE 2930+PEEK(1571)*10:FOR I=0
TO 7:READ WERT:POKE 26640+I,WERT:NEXT I:
POKE 708,58+32*PEEK(1571)                           <SH>
2990 POKE 1582,PEEK(1571):RETURN                    <HZ>
3000 RESTORE 3000                                   <HW>
3010 FOR I=1 TO 8                                   <AF>
3020 READ WERT:SOUND 0,WERT,10,10:READ W
ERT:FOR J=0 TO WERT:NEXT J:SOUND 0,0,0,0
:READ WERT:FOR J=0 TO WERT:NEXT J                   <ZH>
3030 NEXT I:RETURN                                  <DS>
3040 DATA▲99,10,5,99,10,5                           <KH>
3050 DATA▲99,15,5,120,20,15                         <XX>
3060 DATA▲120,15,10,130,20,15                       <QT>
3070 DATA▲130,20,15,140,50,5                        <TK>
3100 POKE 34816,112:POKE 34817,78:POKE 3
4818,0:POKE 34819,144:FOR I=0 TO 44:POKE
 I+34820,14:NEXT I                                  <SF>
3110 POKE 34865,65:POKE 34866,0:POKE 348
67,105                                              <NC>
3120 COLOR 2:PLOT 47,45:DRAWTO 47,0:DRAW
TO 37,0:POSITION 37,45:POKE 765,2:XIO 18
,#6,0,0,"S:"                                        <UH>
3130 PLOT 122,45:DRAWTO 122,0:DRAWTO 112
,0:POSITION 112,45:XIO 18,#6,0,0,"S:"    <WF>
3140 PLOT 48,0:DRAWTO 111,0:PLOT 48,1:DR
AWTO 111,1:PLOT 48,44:DRAWTO 111,44:PLOT
 48,45:DRAWTO 111,45                                <VG>
3150 COLOR 1:FOR I=0 TO 63:FOR J=0 TO 41 <JG>
3160 IF PEEK(28922+I+J*80)=2 THEN PLOT I
+48,J+2                                             <SX>
3170 NEXT J:NEXT I                                  <WV>
3190 RESTORE 3200                                   <KD>
3200 DATA▲104,162,0,138,157,0,139,157,0,
141,232,208,247,96                                  <ZK>
3210 FOR I=34868 TO I+13:READ WERT:POKE
I,WERT:NEXT I                                       <AP>
3220 FOR I=34882 TO I+73:READ WERT:POKE
I,WERT:NEXT I                                       <DM>
3230 FOR I=35072 TO I+104:READ WERT:POKE
 I,WERT:NEXT I                                      <OL>
3290 RETURN                                         <QA>
3300 DATA▲169,152,56,229,207,170,165,206
,201,4,144,1,202,165,20,74,74,176,2,162,
0,142,1,208                                         <UV>
3310 DATA▲173,40,6,24,105,177,170,169,0,
157,0,141,165,205,141,40,6,24,105,177,17
0,165,204,201,10,144,4,232                          <VC>
3320 DATA▲238,40,6,169,1,157,0,141,162,0
,32,0,137,232,224,4,144,248                         <UR>
3330 DATA▲238,26,2,96                               <SQ>
3350 DATA▲189,20,6,56,233,112,141,42,6,1
89,16,6,56,233,250,141,41,6,176,3,206,42
,6,169,0,141,43,6,173,42,6                          <XF>
3360 DATA▲208,7,173,41,6,201,80,144,19,1
73,41,6,56,233,80,141,41,6,176,3,206,42,
6,238,43,6,16,225                                   <SI>
3370 DATA▲169,95,24,109,41,6,157,4,208,1
89,47,6,105,177,168,185,0,139,93,51,6,15
3,0,139,173,43,6,157,47,6                           <OY>
3380 DATA▲189,47,6,105,177,168,185,0,139
,93,51,6,153,0,139,96                               <AR>
3400 POKE 40593,PEEK(40593)+1:IF PEEK(40
593)=35 THEN POKE 40593,33                          <QS>
3410 WERT=LAST(PEEK(40593)-33):A=PEEK(40
593)                                                <BH>
3420 J=HIGH(PEEK(40593)-33)                         <NE>
3430 FOR I=5 TO 0 STEP -1                           <VE>
3440 POKE 40516-I,INT(WERT/10^I)+32:POKE
 40593,A*(I=INT(I/2)*2)+9*(I<>INT(I/2)*2
)                                                   <NK>
3450 POKE 40556-I,INT(J/10^I)+32                    <DU>
3460 WERT=WERT-((PEEK(40516-I)-32)*10^I) <SU>
3470 J=J-((PEEK(40556-I)-32)*10^I)                  <CG>
3480 NEXT I:POKE 40593,A                            <BV>
3490 IF PEEK(53279)<>7 THEN 3490                    <GA>
3500 GOTO 150                                       <PM>
```

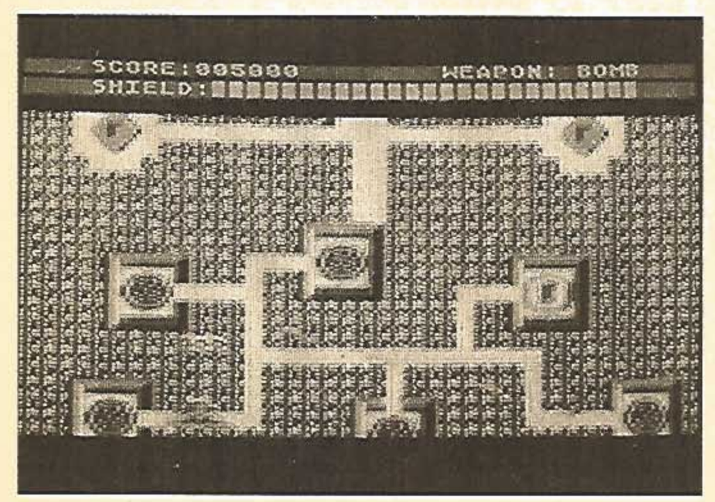
»Wandering« (Schluß)

Kampf um den Kriegsplaneten

Nur mutige und reaktionsschnelle Piloten überleben den mörderischen Flug über Bellum 2.

Der große Krieg zwischen Batarianern und ihrem kriegerischen Nachbarvolk, den Comaden, dauerte schon 10 Jahre. Denis war nicht bereit, sein Leben irgendwo sinnlos zu vergeuden. Er wollte dem Krieg ein Ende setzen.

Schon lange munkelte man von einem riesigen Kriegsplaneten. Dorthin hatten sich die sieben Mächtigen der Comaden zurückgezogen. Wo sich dieser Planet befand, wußte niemand, genau wie niemand sagen konnte, ob das Gerücht der Wahrheit entsprach oder eben nur ein Gerücht war. Denis glaubte dem Gerücht.



»Arax«: Der Kampf beginnt

Das sechs Planeten umfassende System um die kleine, blauweiße Sonne Bellum wurde allgemein als Heimat der Comaden angesehen. Dort vermutete Denis daher auch den Kriegsplaneten. Bevor er an seinem ersten Einsatzort angekommen wäre, desertierte er mit seinem Raumgleiter und nahm Kurs auf das Bellum-System. Dort erwartete ihn tatsächlich ein waffenstarrender Planet: Bellum 2. Aber war es der, von dem die Gerüchte sprachen?

Hier beginnt Ihre Aufgabe. Sie fliegen an Denis Stelle über den Planeten, um sein Geheimnis zu enträtseln. Dabei müssen Sie sich vor den Verteidigungsbasen in acht nehmen, die Sie unaufhörlich beschießen. Die Arax besitzt zwar einen Schutzschirm, doch mehr als fünf Treffer hält sie nicht aus. Die Belastung des Schildes macht sich auch farblich bemerkbar. Bei jedem Treffer wird der sonst dunkle

Steckbrief	
Programm:	Arax
Sprache:	Maschinensprache
Eingabehilfe:	AMPEL
Datenträger:	Diskette, Kassette

Schutzschirm heller und wenn er rötlich glüht, ist der nächste Treffer tödlich.

Sie sind natürlich nicht wehrlos, denn die Arax verfügt über Bomben und eine Schnellfeuerwaffe. Die Bomben sind zwar schwieriger einzusetzen, dafür aber wirkungsvoller. Beim Abwurf schlagen sie genau dort ein, wo das Fadenkreuz vorher zu sehen war. Die Waffe wechselt man durch Druck auf die <SPACE>-Taste.

Wenn Sie das Spiel starten, erscheint erst ein Titelbild. Wenn Sie die <START>-Taste oder den Feuerknof des Joysticks in Port 1 drücken, beginnt das Spiel.

Arax besitzt auch eine Pause-Funktion, damit Sie sich während des Spiels kurz erholen können. Mit der <ESC>-Taste frieren Sie alle Aktionen ein, bis Sie wieder den Feuerknopf drücken.

Im Spiel erhalten Sie für jeden zerstörten Teil einer Basis 100 Punkte und für eine komplett zerstörte Basis sogar 1000 Punkte.

(Martin Horst/gn/hf)

```
Programmname :ARAX.COM
Länge        :17992 Bytes
-----------------------------------

0000:FF FF D8 6C 5F A7 20 24 <6F>
0008:44 32 16 16 16 16 16 16 <24>
0010:16 16 51 52 52 52 52 52 <B8>
0018:52 52 53 16 51 52 52 52 <4B>
0020:52 52 52 52 52 52 52 52 <FB>
0028:52 52 52 52 52 52 52 52 <23>
0030:53 16 51 52 52 52 52 52 <59>
0038:52 53 54 05 55 05 55 05 <5D>
0040:55 05 56 16 54 51 52 52 <32>
0048:52 52 52 52 52 52 52 52 <03>
0050:52 52 52 52 52 52 52 53 <0C>
0058:56 16 54 05 55 05 55 05 <D1>
0060:55 56 54 05 05 4C 05 4C <A6>
0068:05 05 56 16 54 54 51 55 <3F>
0070:4C 55 4C 55 4C 55 4C 55 <EF>
0078:4C 55 4C 55 4C 55 53 56 <F4>
0080:56 16 54 05 05 4C 05 4C <BB>
0088:05 56 54 05 55 05 4C 05 <D7>
0090:55 05 56 16 54 54 55 51 <74>
0098:52 52 52 52 52 52 52 52 <6E>
00A0:52 52 52 52 52 53 55 56 <84>
00A8:56 16 54 05 55 05 4C 05 <70>
00B0:55 56 54 05 05 4C 05 4C <77>
00B8:05 05 56 16 54 54 4C 54 <65>
00C0:51 52 52 52 52 52 52 52 <1B>
00C8:52 52 52 52 53 56 4C 56 <8E>
00D0:56 16 54 05 05 4C 05 4C <CB>
00D8:05 56 54 05 55 05 4C 05 <28>
00E0:55 05 56 16 54 54 55 54 <67>
00E8:54 51 55 4C 55 4C 55 4C <20>
00F0:55 4C 55 53 56 56 55 56 <12>
00F8:56 16 54 05 55 05 4C 05 <60>
0100:55 56 54 05 05 4C 05 4C <8C>
0108:05 05 56 16 54 54 4C 54 <D8>
0110:54 55 51 52 52 52 52 52 <5F>
0118:52 52 53 55 56 56 4C 56 <68>
0120:56 16 54 05 05 4C 05 4C <5D>
0128:05 56 54 05 55 05 4C 05 <37>
0130:55 05 56 16 54 54 55 54 <1C>
0138:54 4C 54 51 55 4C 55 4C <C1>
0140:55 53 56 4C 56 56 55 56 <09>
0148:56 16 54 05 55 05 4C 05 <0A>
0150:55 56 54 05 05 4C 05 4C <DC>
0158:05 05 56 16 54 54 4C 54 <0A>
0160:54 55 54 55 51 52 52 52 <F7>
0168:53 55 56 55 56 56 4C 56 <FF>
0170:56 16 54 05 05 4C 05 4C <2C>
0178:05 56 54 05 55 05 4C 05 <B7>
0180:55 05 56 16 54 54 55 54 <8D>
0188:54 4C 54 4C 54 3F 40 41 <2F>
0190:56 4C 56 4C 56 56 55 56 <18>
0198:56 16 54 05 55 05 4C 05 <7A>
01A0:55 56 54 05 05 4C 05 4C <6D>
01A8:05 05 56 16 54 54 4C 54 <7B>
01B0:54 55 54 55 54 42 4C 44 <88>
01B8:56 55 56 55 56 56 4C 56 <35>
01C0:56 16 54 05 05 4C 05 4C <FD>
01C8:05 56 54 05 55 05 4C 05 <12>
01D0:55 05 56 16 54 54 55 54 <BD>
01D8:54 4C 54 4C 54 45 46 47 <4A>
01E0:56 4C 56 4C 56 56 55 56 <28>
01E8:56 16 54 05 55 05 4C 05 <2A>
01F0:55 56 54 05 05 4C 05 4C <3C>
01F8:05 05 56 16 54 54 4C 54 <AB>
0200:54 55 54 55 57 58 58 58 <37>
0208:59 55 56 55 56 56 4C 56 <DE>
0210:56 16 54 05 05 4C 05 4C <0A>
0218:05 56 54 05 55 05 4C 05 <61>
0220:55 05 56 16 54 54 55 54 <2A>
0228:54 4C 54 57 55 4C 55 4C <3B>
0230:55 59 56 4C 56 56 55 56 <BD>
0238:56 16 54 05 55 05 4C 05 <1C>
0240:55 56 54 05 05 4C 05 4C <CA>
0248:05 05 56 16 54 54 4C 54 <18>
0250:54 55 57 58 58 58 58 58 <1C>
0258:58 58 59 55 56 56 4C 56 <71>
0260:56 16 54 05 05 4C 05 4C <1A>
0268:05 56 54 05 55 05 4C 05 <71>
0270:55 05 56 16 54 54 55 54 <DC>
0278:54 57 55 4C 55 4C 55 4C <1D>
0280:55 4C 55 59 56 56 55 56 <3E>
0288:56 16 54 05 55 05 4C 05 <4C>
0290:55 56 54 05 05 4C 05 4C <1A>
0298:05 05 56 16 54 54 4C 54 <69>
02A0:57 58 58 58 58 58 58 58 <1B>
02A8:58 58 58 58 59 56 4C 56 <88>
02B0:56 16 54 05 05 4C 05 4C <DF>
02B8:05 56 54 05 55 05 4C 05 <C3>
02C0:55 05 56 16 54 54 55 57 <CE>
02C8:58 58 58 58 58 58 58 58 <82>
02D0:58 58 58 58 58 59 55 56 <27>
02D8:56 16 54 05 55 05 4C 05 <3C>
02E0:55 56 54 05 05 4C 05 4C <2A>
02E8:05 05 56 16 54 54 57 55 <D0>
02F0:4C 55 4C 55 4C 55 4C 55 <6C>
02F8:4C 55 4C 55 4C 55 59 56 <6F>
0300:56 16 54 05 05 4C 05 4C <20>
0308:05 56 54 05 55 05 55 05 <6D>
0310:55 05 56 16 54 57 58 58 <2E>
0318:58 58 58 58 58 58 58 58 <09>
0320:58 58 58 58 58 58 58 59 <22>
0328:56 16 54 05 55 05 55 05 <08>
0330:55 56 57 58 58 58 58 58 <EB>
0338:58 58 59 16 57 58 58 58 <ED>
0340:58 58 58 58 58 05 05 05 <BE>
0348:58 58 58 58 58 58 58 58 <F9>
0350:59 16 57 58 58 58 58 58 <A1>
0358:58 59 16 16 16 16 16 16 <CF>
0360:16 16 16 16 16 16 16 16 <62>
0368:16 16 16 16 08 05 05 05 <66>
0370:08 16 16 16 16 16 16 16 <4F>
0378:16 16 16 16 16 16 16 16 <FD>
0380:16 16 16 16 16 16 16 16 <82>
0388:16 16 16 16 16 16 16 16 <6A>
0390:16 16 16 16 05 05 05 05 <96>
0398:05 16 16 16 16 16 16 16 <15>
03A0:16 16 16 16 16 16 16 16 <A6>
03A8:16 16 16 16 16 16 16 16 <8A>
03B0:16 16 16 16 16 16 16 16 <B6>
03B8:16 16 16 16 08 05 05 05 <D6>
03C0:08 16 16 16 16 16 16 16 <BF>
03C8:16 16 48 49 4A 16 16 16 <C9>
03D0:16 16 16 16 16 16 16 16 <D6>
03D8:16 16 16 16 16 16 16 16 <DE>
03E0:16 16 16 16 05 05 05 05 <E6>
03E8:05 16 16 16 16 16 16 16 <62>
03F0:16 16 4B 4C 4D 16 16 16 <38>
03F8:16 16 16 16 16 16 16 16 <7A>
0400:16 48 49 4A 16 16 16 16 <2C>
0408:16 16 16 16 08 05 05 05 <28>
0410:08 16 16 16 16 16 16 16 <10>
0418:16 16 4E 4F 50 16 16 16 <84>
0420:16 16 92 16 16 16 16 16 <97>
0428:16 4B 4C 4D 16 16 16 16 <77>
0430:16 16 16 16 05 05 05 05 <38>
0438:05 16 16 16 16 16 16 16 <B7>
0440:16 16 16 16 16 16 16 16 <48>
0448:16 16 92 16 16 16 16 16 <DF>
0450:16 4E 4F 50 16 16 16 16 <10>
0458:16 16 16 16 08 05 05 05 <78>
0460:08 16 16 16 16 16 16 16 <61>
0468:16 16 16 16 16 16 16 16 <2E>
0470:16 16 92 92 16 16 16 16 <3F>
0478:16 16 16 16 16 16 16 16 <80>
0480:16 16 16 16 05 63 64 65 <20>
0488:05 16 16 16 16 16 16 16 <07>
0490:16 16 16 16 16 16 16 16 <98>
0498:16 16 95 92 16 16 16 16 <36>
04A0:16 16 16 16 16 16 16 16 <A8>
04A8:16 16 16 16 08 66 4C 68 <FF>
04B0:08 16 51 52 52 53 16 16 <5F>
04B8:16 16 16 16 16 16 16 16 <C0>
04C0:16 16 95 92 92 95 16 16 <28>
04C8:16 16 16 16 16 16 16 16 <D0>
04D0:16 16 16 16 05 69 6A 6B <99>
04D8:05 16 54 4C 4C 56 16 16 <34>
04E0:05 06 05 06 05 06 05 06 <3E>
04E8:05 06 92 95 95 95 92 16 <AC>
04F0:16 16 16 16 16 16 16 16 <78>
04F8:16 16 16 16 08 05 05 05 <18>
0500:08 16 54 4C 4C 56 16 16 <D0>
0508:08 05 05 05 05 05 05 05 <93>
0510:05 05 95 92 95 95 95 92 <55>
0518:16 16 16 16 16 16 16 16 <20>
0520:16 16 16 16 05 05 05 05 <2A>
0528:05 16 57 58 58 59 16 16 <F4>
0530:05 05 05 05 05 05 05 05 <3A>
0538:05 05 95 4C 92 95 92 92 <13>
0540:92 16 16 16 16 16 16 16 <07>
0548:16 16 16 16 08 05 05 05 <6A>
0550:08 16 16 16 16 16 16 16 <53>
0558:08 05 05 05 05 05 05 05 <E3>
0560:05 05 92 95 95 92 95 95 <6C>
0568:92 16 16 16 16 16 16 16 <EF>
0570:16 16 16 16 05 05 05 05 <F9>
0578:05 16 16 16 16 16 16 16 <F9>
0580:05 05 05 05 05 06 05 06 <8F>
0588:05 06 95 92 95 95 92 92 <F6>
0590:92 16 16 16 16 51 52 52 <78>
0598:52 53 16 16 08 05 05 05 <28>
05A0:08 05 06 05 06 05 06 05 <56>
05A8:08 05 05 05 05 16 16 16 <AB>
05B0:16 16 92 95 92 95 95 95 <83>
05B8:92 92 16 16 16 54 4C 4C <3A>
05C0:4C 56 16 16 05 05 05 05 <B5>
05C8:05 05 05 05 05 05 05 05 <D2>
05D0:05 05 05 05 08 16 16 16 <66>
05D8:16 16 92 92 95 92 4C 92 <E2>
05E0:92 95 92 16 16 54 4C 4C <53>
05E8:4C 56 16 16 08 05 05 05 <B5>
05F0:05 05 05 05 05 05 05 05 <FA>
05F8:05 05 05 05 05 16 16 16 <79>
0600:16 16 95 92 95 92 95 92 <FA>
0608:92 95 92 16 16 54 4C 4C <18>
0610:4C 56 16 16 05 05 05 05 <47>
0618:05 05 05 05 05 05 05 05 <24>
0620:05 05 05 05 08 16 16 16 <BB>
0628:16 16 92 95 92 95 92 95 <C3>
0630:92 95 92 16 16 57 58 58 <30>
0638:58 59 16 16 08 05 05 05 <0E>
0640:05 06 05 06 05 08 05 05 <A8>
0648:05 05 05 06 05 16 16 16 <DB>
0650:16 16 92 95 95 92 95 95 <3D>
0658:95 95 95 92 16 16 16 16 <D2>
0660:16 16 16 16 05 05 05 05 <5C>
0668:05 16 16 16 16 05 05 05 <64>
0670:05 08 16 16 16 16 16 16 <2E>
0678:16 16 95 4C 95 92 95 92 <16>
0680:95 92 95 95 16 16 16 16 <1B>
0688:16 16 16 16 08 05 05 05 <AC>
0690:08 16 16 16 16 08 05 05 <2A>
0698:05 05 16 51 52 52 53 16 <C4>
06A0:16 16 92 92 92 95 92 95 <2B>
06A8:92 92 95 95 16 16 16 16 <B9>
06B0:16 16 16 16 05 05 05 05 <BC>
06B8:05 16 16 16 16 05 63 64 <CE>
06C0:65 08 16 54 4C 4C 56 16 <9B>
06C8:16 16 92 95 92 92 95 4C <55>
06D0:95 95 92 95 16 16 16 16 <2B>
06D8:16 16 16 16 08 05 05 05 <FC>
06E0:08 16 16 16 16 08 66 4C <82>
06E8:68 05 16 54 4C 4C 56 16 <36>
06F0:16 16 95 92 95 95 95 92 <97>
06F8:95 95 95 92 16 16 16 16 <33>
0700:16 16 16 16 05 63 64 65 <82>
0708:05 16 16 16 16 05 69 6A <23>
0710:6B 08 16 57 58 58 59 16 <F2>
0718:16 16 92 95 92 92 92 92 <C2>
0720:92 92 92 16 16 16 16 16 <B2>
0728:16 16 16 16 08 66 4C 68 <C1>
0730:08 16 16 16 16 08 05 05 <CB>
0738:05 05 16 16 16 16 16 16 <79>
0740:16 16 92 4C 92 92 92 92 <5F>
0748:92 92 92 16 16 16 16 16 <7E>
0750:16 16 16 16 05 69 6A 6B <1C>
0758:05 16 16 16 16 05 05 05 <66>
0760:05 08 16 06 16 06 16 06 <11>
0768:16 16 95 92 95 92 95 92 <00>
0770:95 16 16 16 16 16 16 16 <BD>
0778:16 16 16 16 08 05 05 05 <9E>
0780:08 16 16 16 16 08 05 05 <1C>
0788:05 05 05 05 05 05 05 08 <99>
```

```
0790:16 16 95 92 95 95 95 95 <A8>
0798:95 16 16 16 16 16 16 16 <62>
07A0:16 16 16 16 05 05 05 05 <8E>
07A8:05 16 16 16 16 05 05 05 <96>
07B0:05 05 05 05 05 05 05 05 <BE>
07B8:16 16 92 95 92 92 16 16 <EF>
07C0:16 16 16 16 16 16 16 16 <AE>
07C8:16 16 16 16 08 05 05 05 <EE>
07D0:08 16 16 16 16 08 05 05 <6C>
07D8:05 05 05 05 05 05 05 08 <E9>
07E0:16 16 92 92 16 16 16 16 <24>
07E8:16 16 16 16 16 16 16 16 <72>
07F0:16 16 16 16 05 05 05 05 <7E>
07F8:05 16 16 16 16 16 16 16 <7D>
0800:16 16 16 08 63 64 65 05 <5C>
0808:16 16 16 16 16 16 16 16 <16>
0810:16 16 16 16 16 16 16 16 <1E>
0818:16 16 16 16 08 05 05 05 <40>
0820:08 16 16 16 16 16 16 16 <29>
0828:16 16 16 05 66 4C 68 08 <15>
0830:16 16 16 16 16 16 16 16 <3E>
0838:16 16 16 16 16 16 16 16 <48>
0840:16 16 16 16 05 05 05 05 <50>
0848:05 16 16 16 51 52 52 52 <4A>
0850:53 16 16 08 69 6A 6B 05 <21>
0858:16 16 16 16 16 16 16 51 <A3>
0860:52 52 52 53 16 16 16 16 <78>
0868:16 16 16 16 08 05 05 05 <10>
0870:08 16 16 16 54 4C 4C 4C <DC>
0878:56 16 16 05 05 05 05 08 <9A>
0880:16 16 16 16 16 16 16 54 <CE>
0888:4C 4C 4C 56 16 16 16 16 <BB>
0890:16 16 16 16 05 63 64 65 <38>
0898:05 16 16 16 54 4C 4C 4C <89>
08A0:56 16 16 08 05 05 05 05 <EF>
08A8:05 05 16 16 16 16 16 54 <19>
08B0:4C 4C 4C 56 16 16 16 16 <13>
08B8:16 16 16 16 08 66 4C 68 <57>
08C0:08 16 16 16 54 4C 4C 4C <2E>
08C8:56 16 16 05 05 05 05 05 <67>
08D0:05 05 16 16 16 16 16 54 <51>
08D8:4C 4C 4C 56 16 16 16 16 <CB>
08E0:16 16 16 16 05 69 6A 6B <71>
08E8:05 16 16 16 57 58 58 58 <05>
08F0:59 16 16 08 05 05 05 05 <C0>
08F8:05 05 16 16 16 16 16 57 <74>
0900:58 58 58 59 16 16 16 16 <EF>
0908:16 16 16 16 08 05 05 05 <32>
0910:08 16 16 16 16 16 16 16 <12>
0918:16 16 16 16 06 16 06 16 <89>
0920:06 16 16 16 16 16 16 16 <28>
0928:16 16 16 16 16 16 16 16 <38>
0930:16 16 16 16 05 05 05 05 <42>
0938:05 16 16 16 16 16 16 16 <C1>
0940:16 16 16 16 16 16 16 16 <4E>
0948:16 16 16 16 16 51 52 52 <F7>
0950:52 52 52 52 52 52 52 53 <BA>
0958:16 16 16 16 08 05 05 05 <82>
0960:08 16 16 16 16 16 16 16 <5B>
0968:16 16 16 16 16 16 16 16 <F9>
0970:16 16 16 16 16 54 05 63 <A2>
0978:64 65 05 63 64 65 05 56 <C1>
0980:16 16 16 16 05 05 05 05 <92>
0988:05 16 16 16 16 16 16 16 <11>
0990:16 16 16 16 16 16 16 16 <9A>
0998:16 16 16 16 16 54 05 66 <C9>
09A0:4C 68 05 66 4C 68 05 56 <5C>
09A8:05 06 05 06 05 05 05 05 <09>
09B0:08 16 16 16 16 16 16 16 <AB>
09B8:16 16 16 16 16 16 16 16 <C2>
09C0:16 16 16 16 16 54 05 69 <F4>
09C8:6A 6B 05 69 6A 6B 05 05 <A1>
09D0:05 05 05 05 05 05 05 05 <E2>
09D8:05 16 16 16 16 16 16 16 <5E>
09E0:16 16 16 16 16 16 16 16 <AA>
09E8:16 16 16 16 16 54 05 63 <9A>
09F0:64 65 05 63 64 65 05 05 <D7>
09F8:05 05 05 05 05 05 05 05 <0A>
0A00:08 16 16 16 16 16 16 16 <EC>
0A08:16 16 16 16 16 16 16 16 <0B>
0A10:16 16 16 16 16 54 05 66 <4A>
0A18:4C 68 05 66 4C 68 05 56 <C7>
0A20:05 06 05 06 05 05 05 05 <84>
0A28:05 16 16 16 16 16 16 16 <73>
0A30:16 16 16 16 16 16 16 16 <44>
0A38:16 16 16 16 16 54 05 69 <65>
0A40:6A 6B 05 69 6A 6B 05 56 <78>
0A48:16 16 16 16 08 05 05 05 <64>
0A50:08 16 16 16 16 16 16 16 <5D>
0A58:16 16 16 16 16 16 16 16 <5C>
0A60:16 16 16 16 16 57 58 58 <FF>
0A68:58 58 58 58 58 58 58 59 <D4>
0A70:16 16 16 16 05 05 05 05 <84>
0A78:05 16 16 16 16 16 16 16 <03>
0A80:16 16 16 16 16 16 16 16 <94>
0A88:16 16 16 16 16 16 16 16 <8C>
0A90:16 16 16 16 16 16 16 16 <A4>
0A98:16 16 16 16 08 05 05 05 <B4>
0AA0:08 16 16 16 16 16 16 16 <8D>
0AA8:16 16 16 16 16 16 16 16 <BC>
0AB0:16 16 51 52 52 52 52 52 <63>
0AB8:52 52 52 52 52 52 52 52 <8A>
0AC0:52 52 52 52 52 52 52 52 <82>
0AC8:52 52 52 52 52 52 52 52 <37>
0AD0:52 52 52 52 52 52 52 52 <52>
0AD8:52 53 54 05 05 05 05 05 <90>
0AE0:05 4C 05 05 05 05 05 05 <86>
0AE8:4C 05 05 05 05 05 05 4C <67>
0AF0:05 05 05 05 05 05 05 4C <4A>
0AF8:05 05 05 05 05 05 4C 05 <9A>
0B00:05 56 57 58 58 58 58 58 <C3>
0B08:58 58 58 58 58 58 58 58 <09>
0B10:58 58 58 58 58 58 58 58 <21>
0B18:58 58 58 58 58 58 58 58 <19>
0B20:58 58 58 58 58 58 58 58 <11>
0B28:58 59 04 04 04 04 04 04 <7D>
0B30:04 04 04 04 04 04 04 04 <46>
0B38:04 04 04 04 04 04 04 04 <4E>
0B40:04 04 04 04 04 04 04 04 <56>
0B48:04 04 04 04 04 04 04 04 <5E>
0B50:04 04 04 04 04 04 04 04 <66>
0B58:04 04 04 04 04 04 04 04 <6E>
0B60:04 04 04 04 04 04 04 04 <76>
0B68:04 04 04 04 04 04 04 04 <FD>
0B70:04 04 04 04 04 04 04 04 <86>
0B78:04 04 04 04 04 04 04 04 <8E>
0B80:04 04 04 04 04 04 75 76 <EB>
0B88:77 04 04 04 04 04 04 04 <58>
0B90:04 04 04 04 5A 5B 5C 04 <67>
0B98:04 04 04 04 04 04 04 04 <AE>
0BA0:04 04 04 04 04 04 04 5A <0C>
0BA8:5B 5C 04 04 04 04 78 4C <71>
0BB0:7A 04 04 04 04 04 04 04 <81>
0BB8:04 04 04 04 5D 4C 5F 04 <71>
0BC0:04 04 04 04 04 04 04 04 <D6>
0BC8:04 04 04 04 04 04 04 5D <37>
0BD0:4C 5F 04 04 04 04 7B 7C <C6>
0BD8:7D 04 04 04 04 04 75 76 <7E>
0BE0:77 04 04 04 60 61 62 04 <41>
0BE8:04 04 04 75 76 77 04 04 <F6>
0BF0:04 04 04 04 04 04 04 60 <60>
0BF8:61 62 04 04 04 04 04 04 <10>
0C00:04 04 04 04 04 04 78 4C <44>
0C08:7A 04 04 04 04 04 04 04 <5B>
0C10:04 04 04 78 4C 7A 04 04 <7B>
0C18:04 04 04 04 04 04 04 04 <30>
0C20:04 04 04 04 04 04 04 04 <38>
0C28:04 04 04 04 04 04 7B 7C <A7>
0C30:7D 04 04 04 04 04 04 04 <84>
0C38:04 04 04 7B 7C 7D 04 04 <6D>
0C40:04 04 04 04 04 04 04 04 <58>
0C48:04 04 04 04 04 04 04 04 <60>
0C50:04 04 04 04 04 04 04 04 <68>
0C58:04 04 04 04 04 04 04 04 <70>
0C60:04 04 04 04 04 04 04 04 <78>
0C68:04 04 04 04 04 04 04 04 <80>
0C70:04 04 04 04 04 04 04 04 <88>
0C78:04 5A 5B 5C 04 04 04 04 <96>
0C80:04 04 04 04 04 04 04 04 <98>
0C88:04 04 04 04 04 04 04 04 <A0>
0C90:04 04 04 04 04 04 04 04 <A8>
0C98:04 04 04 04 04 04 04 04 <B0>
0CA0:04 5D 4C 5F 04 04 04 04 <7D>
0CA8:04 04 04 04 04 04 04 04 <C0>
0CB0:04 04 04 5A 5B 5C 04 04 <3A>
0CB8:04 04 04 04 04 04 04 04 <D0>
0CC0:04 04 04 04 04 04 04 04 <D8>
0CC8:04 60 61 62 04 04 04 04 <69>
0CD0:04 75 76 77 04 04 04 04 <8A>
0CD8:04 04 04 5D 4C 5F 04 04 <36>
0CE0:04 04 04 04 04 04 5A 5B <FC>
0CE8:5C 04 04 04 04 04 04 04 <24>
0CF0:04 04 04 04 04 04 04 04 <06>
0CF8:04 78 4C 7A 04 04 04 04 <8D>
0D00:04 04 04 60 61 62 04 04 <30>
0D08:04 04 04 04 04 04 5D 4C <1C>
0D10:5F 04 04 04 04 04 04 04 <D7>
0D18:04 04 04 04 04 04 04 04 <32>
0D20:04 7B 7C 7D 04 04 04 04 <4E>
0D28:04 04 04 04 04 04 04 04 <42>
0D30:04 04 04 04 04 04 60 61 <5F>
0D38:62 04 04 04 04 04 04 04 <00>
0D40:04 04 04 04 04 04 04 04 <5A>
0D48:04 04 04 04 04 04 04 04 <62>
0D50:04 04 04 04 04 04 04 04 <6A>
0D58:04 04 04 04 04 04 04 04 <72>
0D60:04 04 04 04 04 04 04 04 <7A>
0D68:04 04 04 04 04 04 04 04 <82>
0D70:04 04 04 04 04 04 04 04 <8A>
0D78:04 04 04 04 04 04 04 04 <92>
0D80:04 04 51 52 52 52 52 52 <9B>
0D88:52 52 52 52 52 52 52 52 <98>
0D90:52 52 52 52 52 52 52 52 <A0>
0D98:52 52 52 52 52 52 52 52 <A8>
0DA0:52 52 52 52 52 52 52 52 <90>
0DA8:52 53 54 05 05 05 05 05 <A6>
0DB0:05 05 05 4C 05 05 05 05 <2F>
0DB8:05 05 05 05 4C 05 05 05 <04>
0DC0:05 05 05 05 4C 05 05 05 <0B>
0DC8:05 05 05 05 4C 05 05 05 <15>
0DD0:05 56 57 58 58 58 58 58 <4F>
0DD8:58 58 58 58 58 58 58 58 <5C>
0DE0:58 58 58 58 58 58 58 58 <55>
0DE8:58 58 58 58 58 58 58 58 <AD>
0DF0:58 58 58 58 58 58 58 58 <C5>
0DF8:58 59 16 16 16 16 16 16 <C3>
0E00:16 16 16 16 16 16 16 16 <0B>
0E08:16 16 16 16 16 16 16 16 <22>
0E10:16 16 16 16 16 16 16 16 <1A>
0E18:16 16 16 16 16 16 16 16 <F3>
0E20:16 16 92 92 16 16 16 16 <83>
0E28:16 16 16 16 16 16 16 16 <44>
0E30:16 16 16 16 16 16 16 16 <3A>
0E38:16 16 16 16 16 16 16 16 <54>
0E40:16 16 16 16 16 16 16 16 <4C>
0E48:16 16 92 92 92 92 16 16 <61>
0E50:16 16 16 16 16 16 16 16 <5C>
0E58:16 16 16 16 16 16 16 16 <32>
0E60:92 95 92 92 16 16 16 16 <61>
0E68:16 16 16 16 16 16 92 95 <FC>
0E70:92 92 92 4C 95 92 92 92 <0B>
0E78:92 16 16 16 16 16 16 16 <CA>
0E80:92 92 92 92 16 16 16 92 <7C>
0E88:92 95 92 92 92 92 16 16 <83>
0E90:16 16 16 16 92 92 92 92 <E3>
0E98:92 95 92 95 92 92 92 92 <56>
0EA0:92 92 92 92 16 16 92 92 <55>
0EA8:95 92 95 92 92 92 92 92 <F9>
0EB0:95 95 92 95 95 92 92 92 <9C>
0EB8:16 16 92 92 95 92 95 95 <85>
0EC0:92 95 92 95 92 95 92 92 <FA>
0EC8:92 92 92 92 92 92 92 92 <F7>
0ED0:92 92 92 95 92 95 92 92 <2B>
0ED8:92 92 4C 95 95 92 95 95 <A2>
0EE0:92 92 92 92 92 92 92 92 <5F>
0EE8:92 92 95 92 92 95 92 92 <43>
0EF0:95 95 95 95 92 92 92 95 <AF>
0EF8:92 92 92 92 92 95 92 92 <D3>
0F00:92 95 92 92 95 95 92 92 <A8>
0F08:95 92 92 92 92 95 92 92 <40>
0F10:95 95 92 95 92 92 95 95 <44>
0F18:92 95 92 92 92 92 95 95 <AD>
0F20:95 92 92 95 92 92 95 92 <62>
0F28:92 95 95 92 92 95 92 92 <C0>
0F30:4C 92 92 92 95 92 92 92 <C8>
0F38:92 92 95 92 92 4C 95 92 <D4>
0F40:92 95 92 92 4C 95 95 95 <24>
0F48:92 92 95 92 92 95 95 92 <EB>
0F50:95 95 92 92 95 92 92 95 <E6>
0F58:95 92 92 95 92 92 95 4C <55>
0F60:92 92 92 95 95 92 95 95 <36>
0F68:92 95 95 95 95 95 92 92 <C7>
0F70:95 4C 92 92 92 92 95 95 <6E>
0F78:92 92 95 92 92 95 95 92 <B6>
0F80:92 95 92 92 95 95 92 95 <33>
0F88:95 92 92 92 92 92 95 95 <BE>
0F90:95 95 95 92 92 95 95 95 <FF>
0F98:95 95 92 92 95 95 92 92 <07>
0FA0:92 92 92 95 92 92 95 92 <5B>
0FA8:92 92 92 92 92 92 92 92 <1D>
0FB0:92 92 95 92 92 95 92 95 <84>
0FB8:92 95 95 92 92 95 92 92 <30>
0FC0:92 92 95 92 92 92 92 95 <68>
0FC8:92 92 95 4C 95 92 95 92 <17>
0FD0:92 95 4C 95 92 95 92 92 <39>
0FD8:95 92 95 92 95 92 92 95 <5E>
0FE0:4C 92 95 95 92 95 4C 95 <13>
0FE8:92 92 95 92 95 92 4C 95 <CC>
0FF0:92 92 95 92 95 92 92 95 <50>
0FF8:92 92 95 92 95 92 92 95 <48>
1000:92 92 92 95 95 92 95 95 <0C>
1008:92 92 95 95 95 95 95 95 <70>
1010:92 95 95 92 92 92 92 92 <EA>
1018:95 92 92 95 92 92 95 92 <58>
1020:92 95 92 95 92 95 95 92 <9C>
1028:95 95 95 92 95 92 92 95 <4C>
1030:95 92 95 95 95 95 92 95 <E4>
1038:95 92 95 92 92 95 92 92 <6E>
1040:92 92 95 92 92 95 92 92 <E7>
1048:95 95 92 92 95 92 92 95 <D0>
1050:92 95 92 92 95 92 92 92 <63>
1058:92 92 92 95 92 92 95 92 <19>
1060:95 92 92 95 92 92 95 92 <0F>
1068:92 95 95 92 95 95 92 92 <97>
1070:95 92 92 95 92 92 95 92 <1F>
```

»Arax« – Action pur

```
1078:92 95 92 92 95 95 95 92 <2C>
1080:16 16 16 92 92 95 95 92 <BB>
1088:4C 92 92 92 92 16 16 94 <73>
1090:94 92 92 95 4C 92 92 92 <4C>
1098:92 92 95 92 92 16 16 16 <21>
10A0:92 92 16 16 92 92 16 16 <6F>
10A8:16 16 16 16 16 16 16 92 <44>
10B0:92 92 92 16 16 16 16 16 <1D>
10B8:16 94 92 95 95 95 92 16 <98>
10C0:16 92 95 92 16 16 16 16 <57>
10C8:16 92 16 16 16 16 16 16 <C7>
10D0:16 16 16 16 16 16 16 16 <B0>
10D8:92 92 16 16 16 16 16 16 <D5>
10E0:16 16 92 92 92 16 16 16 <12>
10E8:16 16 16 16 16 16 16 16 <07>
10F0:16 16 16 16 16 16 16 16 <0F>
10F8:16 16 16 16 16 16 16 16 <16>
1100:16 16 16 16 16 16 16 16 <20>
1108:16 16 16 16 16 16 16 16 <28>
1110:16 16 16 16 16 16 16 16 <F1>
1118:16 16 16 16 16 16 16 16 <38>
1120:16 16 16 16 16 16 16 16 <40>
1128:16 16 16 16 16 16 16 16 <46>
1130:16 16 16 16 16 16 16 16 <4E>
1138:16 16 16 16 16 16 16 16 <56>
1140:16 16 05 05 05 05 05 05 <2F>
1148:05 05 05 05 05 05 05 51 <B6>
1150:52 52 52 52 52 52 52 52 <E9>
1158:52 52 52 53 05 05 05 05 <62>
1160:05 05 05 05 05 05 05 05 <82>
1168:05 05 05 05 05 05 05 05 <8A>
1170:05 05 05 05 05 05 05 54 <E1>
1178:4C 4C 4C 4C 4C 4C 4C 4C <88>
1180:4C 4C 4C 56 05 05 05 05 <F5>
1188:05 05 05 05 05 05 05 05 <AA>
1190:05 05 05 05 05 05 05 05 <B2>
1198:05 05 05 05 04 04 04 54 <FB>
11A0:4C 4C 4C 4C 4C 4C 4C 4C <A0>
11A8:4C 4C 4C 56 04 04 04 05 <DF>
11B0:05 05 05 05 05 05 05 05 <D2>
11B8:05 05 05 05 05 05 05 05 <DA>
11C0:05 05 05 05 04 05 05 54 <29>
11C8:4C 4C 4C 4C 4C 4C 4C 4C <58>
11D0:4C 4C 4C 56 05 05 04 05 <E3>
11D8:05 05 05 05 05 05 05 05 <FA>
11E0:05 05 05 05 05 05 05 05 <01>
11E8:05 05 05 05 04 05 05 57 <53>
11F0:58 58 58 58 58 58 58 58 <BD>
11F8:58 58 58 59 05 05 04 05 <26>
1200:05 05 05 05 05 05 05 05 <24>
1208:05 05 05 05 05 05 05 05 <2C>
1210:05 05 05 05 04 05 05 05 <2C>
1218:05 05 05 05 05 05 05 05 <1C>
1220:05 05 05 05 05 05 04 05 <42>
1228:05 05 05 05 05 05 05 05 <4C>
1230:05 05 05 05 05 04 04 04 <4D>
1238:04 04 04 04 04 04 04 04 <4C>
1240:04 04 04 04 04 04 04 04 <64>
1248:04 04 04 04 04 04 04 04 <5C>
1250:04 04 04 04 04 04 05 05 <77>
1258:05 05 05 05 05 04 05 05 <38>
1260:05 05 05 05 04 05 05 05 <7C>
1268:05 05 05 05 05 04 05 05 <88>
1270:05 05 05 05 05 05 04 05 <92>
1278:05 05 05 05 05 04 05 05 <98>
1280:05 05 05 51 52 04 52 53 <B0>
1288:05 05 05 05 04 05 05 05 <A4>
1290:05 05 05 05 04 04 04 05 <A6>
1298:05 05 05 05 05 05 05 05 <7A>
12A0:05 05 05 51 52 04 52 53 <D0>
12A8:05 05 05 54 2D 2E 2F 56 <48>
12B0:05 05 05 51 52 53 05 05 <35>
12B8:05 05 05 04 5A 5B 5C 04 <7C>
12C0:05 05 05 05 05 51 52 53 <FE>
12C8:05 05 05 54 2D 2E 2F 56 <69>
12D0:05 05 05 54 30 31 32 56 <97>
12D8:05 05 05 54 4C 56 05 05 <29>
12E0:05 05 05 04 5D 5E 5F 04 <CB>
12E8:05 05 05 05 05 54 4C 56 <28>
12F0:05 05 05 54 30 31 32 56 <BA>
12F8:05 05 05 54 33 34 35 56 <CC>
1300:05 05 05 57 58 59 05 05 <2B>
1308:05 05 05 04 60 61 62 04 <22>
1310:05 05 05 05 05 57 58 59 <76>
1318:05 05 05 54 33 34 35 56 <0A>
1320:05 05 05 57 58 58 58 59 <39>
1328:05 05 05 05 05 05 05 05 <4E>
1330:05 05 05 05 04 04 04 05 <48>
1338:05 05 05 05 05 05 04 05 <5E>
1340:05 05 05 57 58 58 58 59 <5A>
1348:05 05 05 05 05 05 05 05 <6E>
1350:05 05 05 05 05 05 05 05 <76>
1358:05 05 05 05 05 05 05 05 <FD>
1360:05 05 05 05 05 05 05 05 <86>
1368:05 05 05 05 05 05 05 05 <8E>
1370:05 05 17 17 17 17 17 17 <08>
1378:17 17 17 17 17 17 17 17 <9E>
1380:17 17 17 17 05 05 05 17 <A9>
1388:17 17 17 17 17 17 17 17 <8E>
1390:17 17 17 17 17 17 17 17 <B6>
1398:17 17 05 05 05 55 05 55 <DD>
13A0:05 05 05 05 05 05 05 05 <C6>
13A8:05 55 05 55 05 05 05 55 <37>
13B0:05 55 05 05 05 05 05 05 <EA>
13B8:05 05 05 55 05 55 05 05 <25>
13C0:05 05 05 55 05 2D 2E 2F <F8>
13C8:05 55 05 55 05 55 05 55 <57>
13D0:05 24 25 26 05 55 05 24 <F4>
13D8:25 26 05 55 05 55 05 55 <5B>
13E0:05 55 05 2D 2E 2F 05 55 <CE>
13E8:05 05 05 05 05 30 4C 32 <74>
13F0:05 05 05 05 05 05 05 05 <16>
13F8:05 27 28 29 05 05 05 27 <5F>
1400:28 29 05 05 05 05 05 05 <82>
1408:05 05 05 30 4C 32 05 05 <C9>
1410:05 05 05 55 05 33 34 35 <74>
1418:05 55 05 05 08 05 05 55 <BC>
1420:05 2A 2B 2C 05 55 05 2A <2F>
1428:2B 2C 05 55 05 05 08 05 <38>
1430:05 55 05 33 34 35 05 55 <D9>
1438:05 05 05 05 05 55 05 55 <F1>
1440:05 05 05 05 05 05 05 05 <68>
1448:05 55 05 55 05 05 05 55 <99>
1450:05 55 05 05 05 05 05 05 <4C>
1458:05 05 05 55 05 55 05 05 <C6>
1460:05 05 05 55 05 05 05 05 <7D>
1468:05 4C 05 05 08 05 05 4C <C1>
1470:05 05 05 05 05 55 05 05 <D9>
1478:05 05 05 4C 05 05 08 05 <1B>
1480:05 4C 05 05 05 05 05 55 <CA>
1488:05 05 05 05 05 05 08 05 <B6>
1490:05 05 05 05 05 05 05 05 <B8>
1498:05 05 08 05 05 05 05 05 <21>
14A0:08 05 05 05 05 05 05 05 <4A>
14A8:05 05 05 05 08 05 05 05 <E8>
14B0:05 05 05 55 05 05 05 05 <CD>
14B8:05 55 05 05 08 05 05 55 <5B>
14C0:05 05 05 05 05 4C 05 05 <06>
14C8:05 05 05 55 05 05 08 05 <FB>
14D0:05 55 05 05 05 05 05 55 <1C>
14D8:05 05 05 05 05 05 08 05 <06>
14E0:05 05 05 63 64 65 05 05 <5A>
14E8:05 05 08 05 05 05 05 05 <70>
14F0:08 05 05 05 05 63 64 65 <2D>
14F8:05 05 05 05 08 05 05 05 <38>
1500:05 05 05 55 05 05 05 05 <1B>
1508:05 55 05 66 4C 68 05 55 <1F>
1510:05 05 05 05 05 4C 05 05 <53>
1518:05 05 05 55 05 66 4C 68 <BC>
1520:05 55 05 05 05 05 05 55 <AE>
1528:05 05 05 05 05 05 08 05 <58>
1530:05 05 05 69 6A 6B 05 05 <4D>
1538:05 05 08 05 05 05 05 05 <C2>
1540:08 05 05 05 05 69 6A 6B <AE>
1548:05 05 05 05 08 05 05 05 <8A>
1550:05 05 05 55 05 05 05 05 <6F>
1558:05 55 05 05 05 05 05 55 <E6>
1560:05 05 05 05 05 4C 05 05 <A3>
1568:05 05 05 55 05 05 05 05 <97>
1570:05 55 05 05 05 05 05 55 <FE>
1578:05 05 05 05 05 05 08 05 <A8>
1580:05 05 05 55 05 55 05 05 <E0>
1588:05 05 08 05 05 05 05 05 <0F>
1590:08 05 05 05 05 55 05 55 <AD>
1598:05 05 05 05 08 05 05 05 <DA>
15A0:05 05 05 55 05 63 64 65 <57>
15A8:05 55 05 2D 2E 2F 05 55 <A7>
15B0:05 63 64 65 05 4C 05 63 <BF>
15B8:64 65 05 55 05 2D 2E 2F <38>
15C0:05 55 05 63 64 65 05 55 <71>
15C8:05 05 05 05 05 66 4C 68 <65>
15D0:05 05 05 30 4C 32 05 05 <8C>
15D8:05 66 4C 68 05 05 05 66 <BA>
15E0:4C 68 05 05 05 30 4C 32 <AC>
15E8:05 05 05 66 4C 68 05 05 <EB>
15F0:05 05 05 55 05 69 6A 6B <CB>
15F8:05 55 05 33 34 35 05 55 <9F>
1600:05 69 6A 6B 05 55 05 69 <BD>
1608:6A 6B 05 55 05 33 34 35 <8C>
1610:05 55 05 69 6A 6B 05 55 <67>
1618:05 05 05 06 05 05 05 05 <54>
1620:05 05 05 55 05 55 05 05 <92>
1628:05 05 05 05 05 06 05 05 <58>
1630:05 05 05 05 05 55 05 55 <ED>
1638:05 05 05 05 05 05 05 06 <65>
1640:05 05 05 05 05 05 05 05 <6C>
1648:05 05 05 05 05 05 05 05 <74>
1650:05 05 05 05 05 05 05 05 <3C>
1658:05 05 05 05 05 05 05 05 <84>
1660:05 05 05 05 05 05 05 05 <8C>
1668:05 05 05 05 06 06 06 05 <A2>
1670:05 05 06 06 06 05 05 05 <D4>
1678:06 06 06 05 05 05 06 06 <88>
1680:06 05 05 05 06 06 06 05 <3B>
1688:05 05 06 06 06 05 05 05 <EC>
1690:06 06 05 05 05 05 05 05 <3D>
1698:05 05 05 05 05 05 05 05 <C4>
16A0:05 05 05 05 05 05 05 05 <CC>
16A8:05 05 05 05 05 05 05 05 <D4>
16B0:05 05 05 05 05 05 05 05 <DC>
16B8:05 05 05 06 05 06 05 06 <F9>
16C0:05 06 05 06 05 06 05 06 <42>
16C8:05 06 05 06 05 08 05 06 <52>
16D0:05 06 05 06 05 06 05 06 <12>
16D8:05 06 05 06 05 06 05 06 <59>
16E0:05 06 17 17 17 17 17 17 <BE>
16E8:17 17 17 17 17 17 17 17 <13>
16F0:17 17 17 05 05 05 05 05 <EB>
16F8:17 17 17 17 17 17 17 17 <24>
1700:17 17 17 17 17 17 17 17 <0D>
1708:17 17 92 92 92 92 92 92 <82>
1710:92 92 92 92 92 92 92 92 <9F>
1718:92 92 92 05 05 08 05 05 <8D>
1720:92 92 92 92 92 92 92 92 <8E>
1728:92 92 92 92 92 92 95 92 <8C>
1730:92 92 92 95 95 4C 4C 4C <DD>
1738:95 95 95 95 95 92 95 95 <64>
1740:95 95 92 05 05 05 05 05 <AB>
1748:93 95 95 92 95 92 95 92 <62>
1750:95 95 95 92 4C 4C 4C 95 <A9>
1758:92 92 95 95 95 4C 4C 4C <12>
1760:92 92 95 92 95 92 95 95 <DD>
1768:95 95 95 05 05 08 05 05 <E0>
1770:92 95 92 92 95 95 95 95 <9A>
1778:95 95 92 95 4C 4C 4C 95 <D1>
1780:95 92 92 95 95 4C 4C 4C <2B>
1788:95 95 95 95 95 92 95 92 <10>
1790:92 95 95 05 05 05 05 05 <33>
1798:93 95 92 95 95 95 92 95 <AC>
17A0:92 92 92 95 4C 4C 4C 92 <6D>
17A8:95 92 92 95 95 95 95 95 <E0>
17B0:95 95 95 92 95 95 95 92 <C4>
17B8:92 92 92 05 05 08 05 05 <ED>
17C0:92 92 95 95 92 92 95 92 <8B>
17C8:92 92 95 95 92 95 95 92 <C6>
17D0:95 92 05 05 05 05 05 05 <2A>
17D8:05 05 05 05 05 05 05 05 <06>
17E0:05 05 05 05 05 05 05 05 <0E>
17E8:05 05 05 05 05 05 05 05 <16>
17F0:05 05 05 05 05 05 05 05 <1E>
17F8:05 05 06 05 06 05 06 06 <3B>
1800:05 06 05 06 05 06 05 06 <85>
1808:05 06 05 06 05 08 05 06 <95>
1810:05 06 05 06 05 06 05 06 <95>
1818:05 06 05 06 05 06 05 06 <9D>
1820:05 06 05 05 05 05 05 05 <90>
1828:05 05 05 05 05 05 05 05 <58>
1830:05 05 05 05 05 05 05 05 <60>
1838:05 05 05 05 05 05 05 05 <68>
1840:05 05 05 05 05 05 05 05 <70>
1848:05 05 17 17 17 17 17 17 <DA>
1850:17 17 17 17 17 17 17 17 <80>
1858:17 17 17 05 05 08 05 05 <64>
1860:17 17 17 17 17 17 17 17 <90>
1868:17 17 17 17 17 17 17 17 <98>
1870:17 17 05 05 05 05 05 05 <2E>
1878:05 05 05 05 05 05 05 05 <A8>
1880:05 05 05 05 05 05 05 05 <B0>
1888:05 05 05 05 05 05 05 05 <B8>
1890:05 05 05 05 05 05 05 05 <C0>
1898:05 05 05 06 05 06 05 06 <DD>
18A0:05 06 05 06 05 06 05 06 <26>
18A8:05 06 05 06 05 08 05 06 <36>
18B0:05 06 05 06 05 06 05 06 <36>
18B8:05 06 05 06 05 06 05 06 <3E>
18C0:05 06 05 05 05 05 05 05 <31>
18C8:05 05 05 05 05 05 05 05 <F8>
18D0:05 05 05 05 05 05 05 05 <FF>
18D8:05 05 05 05 05 05 05 05 <08>
18E0:05 05 05 05 05 05 05 05 <10>
18E8:05 05 95 95 95 92 92 95 <60>
18F0:95 92 95 95 95 92 95 92 <EB>
18F8:92 92 92 95 95 95 92 95 <0A>
1900:95 92 95 92 95 95 95 95 <F4>
1908:95 92 95 92 92 95 92 92 <8C>
1910:95 92 92 92 95 95 92 92 <3C>
1918:95 95 95 92 92 95 95 95 <44>
1920:95 92 95 92 95 95 95 92 <62>
1928:92 92 95 95 95 95 92 95 <38>
1930:95 92 95 92 92 95 92 95 <67>
1938:95 92 95 95 92 95 95 92 <93>
1940:95 92 16 16 16 16 16 92 <0C>
1948:95 95 95 92 92 95 95 95 <94>
1950:95 95 92 92 92 92 92 95 <99>
1958:95 95 92 95 92 92 95 95 <C7>
1960:95 92 95 95 95 92 95 95 <78>
1968:92 16 16 75 76 77 16 16 <43>
1970:95 95 95 16 16 16 92 95 <47>
```

```
1978:95 92 16 16 16 16 92 95 <BF>
1980:92 95 92 95 95 92 95 92 <76>
1988:95 92 95 95 95 95 92 92 <53>
1990:16 16 16 78 4C 7A 16 16 <26>
1998:92 92 16 16 16 16 16 92 <23>
19A0:95 16 09 0A 0B 16 16 95 <D6>
19A8:95 95 95 92 95 95 92 92 <C2>
19B0:95 92 95 95 95 95 92 95 <2E>
19B8:92 16 16 7B 7C 7D 16 16 <BB>
19C0:92 92 16 5A 5B 5C 16 16 <12>
19C8:92 16 0C 4C 0E 16 16 16 <96>
19D0:16 92 92 92 92 95 95 95 <AE>
19D8:92 92 95 92 92 92 92 92 <31>
19E0:92 16 16 16 04 16 16 16 <BF>
19E8:95 16 16 5D 4C 5F 16 16 <14>
19F0:95 16 0F 10 11 16 16 16 <78>
19F8:16 95 95 92 95 92 92 95 <A9>
1A00:95 92 92 95 92 92 95 95 <A0>
1A08:16 16 16 16 04 16 16 92 <17>
1A10:92 16 16 60 61 62 16 16 <30>
1A18:95 16 16 0F 16 09 0A 0B <BB>
1A20:16 16 16 95 95 95 92 92 <A9>
1A28:95 92 95 95 95 95 92 92 <B3>
1A30:16 04 04 04 04 16 95 92 <64>
1A38:95 95 16 16 04 16 16 92 <2E>
1A40:92 92 16 16 16 0C 4C 0E <4D>
1A48:16 16 16 95 92 92 95 92 <32>
1A50:95 92 95 95 95 92 92 16 <09>
1A58:16 04 16 16 16 16 92 92 <7A>
1A60:92 92 92 16 04 16 95 95 <2E>
1A68:92 92 92 92 16 0F 10 11 <D3>
1A70:16 16 16 16 16 92 95 95 <10>
1A78:95 92 95 92 92 92 16 16 <A0>
1A80:16 04 16 16 16 16 92 95 <65>
1A88:95 95 92 16 04 16 95 95 <48>
1A90:95 95 95 16 16 16 16 16 <B3>
1A98:09 0A 0B 16 16 92 92 95 <45>
1AA0:95 92 92 92 92 95 16 16 <1E>
1AA8:75 76 77 16 16 16 16 16 <2E>
1AB0:95 95 92 16 04 16 16 92 <16>
1AB8:95 92 16 16 16 16 16 16 <02>
1AC0:0C 4C 0E 16 16 16 92 92 <9F>
1AC8:95 92 92 95 95 95 16 16 <C4>
1AD0:78 4C 7A 04 04 04 04 16 <10>
1AD8:92 92 16 16 04 16 16 92 <54>
1AE0:92 16 16 16 16 16 16 16 <4E>
1AE8:0F 10 11 16 16 16 16 92 <EE>
1AF0:95 92 95 95 95 92 16 16 <6C>
1AF8:7B 7C 7D 16 16 16 04 16 <00>
1B00:16 16 16 3F 40 41 16 16 <C2>
1B08:16 16 16 16 92 92 95 16 <0C>
1B10:16 16 16 09 0A 0B 16 92 <64>
1B18:92 92 95 95 92 95 16 16 <BC>
1B20:16 16 16 16 16 16 04 04 <1C>
1B28:04 04 04 42 4C 44 16 16 <B3>
1B30:16 16 16 92 92 95 92 92 <6E>
1B38:16 16 16 0C 4C 0E 16 16 <3F>
1B40:92 92 92 95 95 92 92 16 <75>
1B48:16 16 16 92 92 16 16 16 <A9>
1B50:16 16 16 45 46 47 16 16 <B7>
1B58:92 95 92 95 92 95 92 92 <58>
1B60:16 16 16 0F 10 11 16 95 <60>
1B68:95 92 92 95 95 92 92 92 <03>
1B70:92 95 95 92 92 92 92 95 <57>
1B78:92 16 16 16 16 16 16 16 <CC>
1B80:95 95 95 95 92 95 95 95 <07>
1B88:92 92 16 16 16 16 95 95 <07>
1B90:92 92 92 95 95 92 92 95 <68>
1B98:92 92 95 95 95 95 92 92 <D0>
1BA0:95 95 95 16 16 16 92 92 <18>
1BA8:95 92 92 95 92 92 95 92 <C1>
1BB0:95 95 92 92 92 95 95 95 <48>
1BB8:95 92 95 95 92 95 95 95 <30>
1BC0:92 95 95 95 92 92 92 95 <77>
1BC8:95 92 92 92 92 95 95 92 <ED>
1BD0:92 95 95 92 16 16 16 95 <2A>
1BD8:95 92 92 92 92 95 95 92 <6E>
1BE0:92 92 95 92 92 92 95 16 <B7>
1BE8:16 16 92 92 95 92 92 92 <B2>
1BF0:95 95 92 92 92 95 95 92 <05>
1BF8:92 92 16 16 16 16 16 16 <43>
1C00:16 16 16 92 92 95 95 92 <42>
1C08:92 92 92 92 95 92 16 16 <4E>
1C10:16 16 16 16 16 95 92 92 <B7>
1C18:16 16 16 16 92 95 95 92 <A3>
1C20:16 16 16 16 16 16 16 16 <58>
1C28:16 16 16 16 16 16 16 92 <DC>
1C30:92 95 95 95 92 16 16 16 <61>
1C38:16 16 16 16 16 16 16 16 <2E>
1C40:16 16 16 16 16 16 16 16 <F7>
1C48:16 16 16 16 16 16 16 16 <80>
1C50:16 16 16 16 16 16 16 92 <04>
1C58:92 92 92 92 16 09 0A 0B <CD>
1C60:16 16 16 16 16 16 16 16 <98>
1C68:48 49 4A 16 16 16 16 16 <EC>
1C70:48 49 4A 16 16 16 16 16 <F4>
1C78:16 16 16 09 0A 0B 16 16 <12>
1C80:95 92 95 92 16 0C 4C 0E <FA>
1C88:16 16 92 95 92 92 16 16 <F5>
1C90:4B 4C 4D 16 16 16 16 16 <97>
1C98:4B 4C 4D 16 16 92 95 92 <0A>
1CA0:95 92 16 0C 4C 0E 16 16 <E7>
1CA8:92 92 92 95 16 0F 10 11 <B7>
1CB0:16 92 92 95 95 92 16 16 <04>
1CB8:4E 4F 50 16 16 16 16 16 <20>
1CC0:4E 4F 50 16 16 92 92 95 <8D>
1CC8:95 92 16 0F 10 11 16 16 <28>
1CD0:95 92 95 92 16 16 16 16 <6A>
1CD8:16 92 95 92 92 95 92 16 <99>
1CE0:16 16 16 16 16 16 16 16 <16>
1CE8:16 16 16 16 95 92 95 92 <7E>
1CF0:95 92 16 16 16 16 16 92 <42>
1CF8:92 92 92 95 92 16 16 16 <98>
1D00:16 92 95 92 95 92 92 16 <DB>
1D08:16 16 16 16 16 16 16 16 <40>
1D10:16 16 16 16 92 92 95 95 <8F>
1D18:92 16 16 16 16 16 16 95 <8E>
1D20:95 92 92 95 92 92 16 16 <BD>
1D28:16 16 16 16 16 16 16 16 <5E>
1D30:16 16 16 16 16 16 16 16 <66>
1D38:16 16 16 16 16 16 16 16 <30>
1D40:16 16 16 16 16 16 92 95 <6F>
1D48:95 92 92 92 95 92 95 16 <79>
1D50:16 16 48 49 4A 16 16 16 <85>
1D58:51 52 52 52 52 52 52 52 <C9>
1D60:52 52 53 16 16 16 48 49 <E6>
1D68:4A 16 16 16 16 16 95 92 <37>
1D70:92 92 92 95 95 92 16 16 <04>
1D78:16 16 4B 4C 4D 16 16 16 <24>
1D80:54 51 52 52 52 52 52 52 <57>
1D88:52 53 56 16 16 16 4B 4C <96>
1D90:4D 16 16 16 16 16 92 95 <DA>
1D98:95 92 92 95 95 16 16 16 <6C>
1DA0:16 16 4E 4F 50 16 16 16 <47>
1DA8:54 54 51 52 52 52 52 52 <AF>
1DB0:53 56 56 16 16 16 4E 4F <CA>
1DB8:50 16 16 16 16 16 16 92 <0B>
1DC0:95 92 92 92 16 16 16 16 <A0>
1DC8:16 16 16 16 16 16 16 16 <01>
1DD0:54 54 54 51 52 52 52 53 <78>
1DD8:56 56 56 16 16 16 16 16 <09>
1DE0:16 16 16 16 16 16 16 16 <18>
1DE8:95 92 95 95 09 0A 0B 16 <D9>
1DF0:16 16 92 92 16 16 16 16 <51>
1DF8:54 54 54 54 4C 4C 4C 56 <AF>
1E00:56 56 56 16 16 16 16 16 <54>
1E08:92 95 92 16 09 0A 0B 16 <A2>
1E10:95 92 95 95 0C 4C 0E 16 <D6>
1E18:16 92 92 92 92 16 16 16 <62>
1E20:54 54 54 54 4C 4C 4C 56 <4C>
1E28:56 56 56 16 16 16 95 92 <84>
1E30:95 92 92 16 0C 4C 0E 16 <6E>
1E38:95 92 95 92 0F 10 11 16 <EF>
1E40:16 92 92 92 92 16 16 16 <3C>
1E48:54 54 54 54 4C 4C 4C 56 <C5>
1E50:56 56 56 16 16 16 92 95 <FC>
1E58:92 95 16 16 0F 10 11 16 <16>
1E60:95 92 92 95 16 16 16 16 <AA>
1E68:16 16 16 16 16 16 16 16 <A4>
1E70:54 54 54 57 58 58 58 59 <15>
1E78:56 56 56 16 16 16 16 16 <CC>
1E80:16 16 16 16 16 16 16 16 <BC>
1E88:92 92 95 95 16 16 16 16 <69>
1E90:16 16 48 49 4A 16 16 16 <C7>
1E98:54 54 57 58 58 58 58 58 <5D>
1EA0:59 56 56 16 16 16 48 49 <8B>
1EA8:4A 16 16 16 16 16 16 92 <F6>
1EB0:95 92 95 95 95 95 16 16 <B4>
1EB8:16 16 4B 4C 4D 16 16 16 <68>
1EC0:54 57 58 58 58 58 58 58 <15>
1EC8:58 59 56 16 16 16 4B 4C <5C>
1ED0:4D 16 16 16 16 95 95 92 <1E>
1ED8:92 92 95 95 92 92 95 95 <60>
1EE0:16 16 4E 4F 50 16 16 16 <88>
1EE8:57 58 58 58 58 58 58 58 <9B>
1EF0:58 58 59 16 16 16 4E 4F <EB>
1EF8:50 16 16 16 92 95 92 95 <9D>
1F00:95 92 95 92 95 95 92 95 <B4>
1F08:92 16 16 16 16 16 16 16 <03>
1F10:16 16 16 16 16 16 16 16 <2C>
1F18:16 16 16 16 16 16 16 16 <52>
1F20:16 16 16 92 92 92 95 95 <73>
1F28:92 92 92 92 95 95 92 95 <9E>
1F30:95 92 92 16 16 16 16 16 <EB>
1F38:16 16 16 16 16 16 16 16 <F5>
1F40:16 16 16 16 16 16 16 16 <FD>
1F48:16 16 92 92 95 92 92 95 <0D>
1F50:95 92 17 17 17 17 17 17 <8C>
1F58:17 17 17 17 17 17 17 17 <96>
1F60:17 17 17 17 17 17 17 17 <9E>
1F68:17 17 17 17 17 17 17 17 <A6>
1F70:17 17 17 17 17 17 17 17 <8E>
1F78:17 17 51 52 52 52 52 52 <21>
1F80:52 52 52 52 52 52 52 52 <9C>
1F88:52 52 52 52 52 52 52 52 <74>
1F90:52 52 52 52 52 52 52 52 <8C>
1F98:52 52 52 52 52 52 52 52 <84>
1FA0:53 05 54 05 55 05 05 05 <F1>
1FA8:55 05 05 05 55 05 05 05 <11>
1FB0:55 05 05 05 55 05 05 55 <69>
1FB8:05 05 05 55 05 05 05 55 <3B>
1FC0:05 05 05 55 05 05 55 05 <24>
1FC8:56 05 54 05 05 05 05 05 <98>
1FD0:05 05 05 05 05 05 05 05 <0E>
1FD8:05 05 05 05 05 05 05 05 <16>
1FE0:05 05 05 05 05 05 05 05 <1E>
1FE8:05 05 05 05 05 05 05 05 <26>
1FF0:56 05 54 05 05 05 6C 6D <D5>
1FF8:6E 06 05 06 05 06 05 06 <38>
2000:05 06 51 52 52 52 52 52 <54>
2008:53 06 05 06 05 06 05 06 <C4>
2010:05 06 6C 6D 6E 05 05 05 <2F>
2018:56 05 54 05 05 05 6F 4C <84>
2020:71 05 05 05 05 05 05 05 <16>
2028:05 05 54 4C 4C 4C 4C 4C <F3>
2030:56 05 05 05 05 05 05 05 <98>
2038:05 05 6F 4C 71 05 05 05 <75>
2040:56 05 54 05 55 05 72 73 <D6>
2048:74 05 05 05 05 05 05 05 <40>
2050:05 05 57 58 58 58 58 58 <DD>
2058:59 05 05 05 05 05 05 05 <C2>
2060:05 05 72 73 74 05 55 05 <4F>
2068:56 05 54 05 05 05 05 05 <33>
2070:05 05 05 05 24 25 26 05 <6C>
2078:05 55 05 05 05 05 05 05 <8C>
2080:05 55 05 05 24 25 26 05 <90>
2088:05 05 05 05 05 05 05 05 <C8>
2090:56 05 54 05 05 05 05 08 <3E>
2098:05 05 05 05 27 28 29 05 <BE>
20A0:05 05 05 05 05 08 05 05 <EC>
20A8:05 05 05 05 27 28 29 05 <CE>
20B0:05 05 05 08 05 05 05 05 <21>
20B8:56 05 54 05 05 05 05 05 <09>
20C0:05 05 05 05 2A 2B 2C 05 <10>
20C8:05 05 05 56 05 05 05 54 <5A>
20D0:05 05 05 05 2A 2B 2C 05 <20>
20D8:05 05 05 05 05 05 05 05 <18>
20E0:56 05 54 05 55 05 05 08 <30>
20E8:05 06 05 06 05 06 05 05 <7C>
20F0:05 05 05 56 24 25 26 54 <4F>
20F8:05 05 05 05 05 06 05 06 <3D>
2100:05 06 05 08 05 05 55 05 <51>
2108:56 05 54 05 05 05 05 05 <BC>
2110:05 05 05 05 05 05 05 05 <52>
2118:05 55 05 56 27 4C 29 54 <48>
2120:05 55 05 05 05 05 05 05 <76>
2128:05 05 05 05 05 05 05 05 <6A>
2130:56 05 54 05 05 05 05 08 <67>
2138:05 05 05 05 05 05 05 05 <7A>
2140:05 05 05 56 2A 2B 2C 54 <F6>
2148:05 05 05 05 05 05 05 05 <8A>
2150:05 05 05 08 05 05 05 05 <C2>
2158:56 05 54 05 05 05 05 05 <2D>
2160:05 05 05 05 05 6C 6D 6E <74>
2168:05 05 05 56 05 05 05 54 <FE>
2170:05 05 05 6C 6D 6E 05 05 <0A>
2178:05 05 05 05 05 05 05 05 <BA>
2180:56 05 54 05 55 05 05 24 <F6>
2188:25 26 05 06 05 6F 4C 71 <CD>
2190:05 05 05 56 24 25 26 54 <F2>
2198:05 05 05 6F 4C 71 05 06 <56>
21A0:05 24 25 26 05 05 55 05 <5F>
21A8:56 05 54 05 05 05 05 27 <FE>
21B0:28 29 05 05 05 72 73 74 <0C>
21B8:05 55 05 56 27 4C 29 54 <A9>
21C0:05 55 05 72 73 74 05 05 <0E>
21C8:05 27 28 29 05 05 05 05 <39>
21D0:56 05 54 05 05 05 05 2A <A9>
21D8:2B 2C 05 05 05 05 05 05 <F6>
21E0:05 05 05 56 2A 2B 2C 54 <94>
21E8:05 05 05 05 05 05 05 05 <2A>
21F0:05 2A 2B 2C 05 05 05 05 <92>
21F8:56 05 54 05 05 05 05 05 <CC>
2200:05 05 05 05 05 05 05 05 <44>
2208:05 05 05 56 05 05 05 54 <B0>
2210:05 05 05 05 05 05 05 05 <54>
2218:05 05 05 05 05 05 05 05 <5C>
2220:56 05 54 05 55 05 05 05 <F8>
2228:05 55 05 05 05 55 05 05 <81>
2230:05 55 05 56 05 08 05 54 <A8>
2238:05 55 05 05 05 05 55 05 <F0>
2240:05 05 55 05 05 05 55 05 <2D>
2248:56 05 57 58 58 58 58 58 <74>
2250:58 58 58 58 58 58 58 58 <6A>
2258:58 58 58 59 05 05 05 57 <FC>
2260:58 58 58 58 58 58 58 58 <9A>
```

»Arax« (Fortsetzung)

```
2268:58 58 58 58 58 58 58 58 <A2>
2270:59 05 17 17 17 17 17 17 <4D>
2278:17 17 17 17 17 17 17 17 <BC>
2280:17 17 17 17 05 08 05 17 <D3>
2288:17 17 17 17 17 17 17 17 <AC>
2290:17 17 17 17 17 17 17 17 <D4>
2298:17 17 05 05 05 05 05 05 <6A>
22A0:05 05 05 05 05 05 05 05 <E4>
22A8:05 05 05 05 05 05 05 05 <EC>
22B0:05 05 05 05 05 05 05 05 <F4>
22B8:05 05 05 05 05 05 05 05 <BC>
22C0:05 05 05 06 05 06 05 06 <19>
22C8:05 06 05 06 05 06 05 06 <61>
22D0:05 06 05 06 05 06 05 06 <69>
22D8:05 06 05 06 05 06 05 06 <71>
22E0:05 06 05 06 05 06 05 06 <79>
22E8:05 06 05 05 05 05 05 05 <6C>
22F0:05 05 05 05 05 05 05 05 <34>
22F8:05 05 05 05 05 05 05 05 <1C>
2300:05 05 05 05 05 05 05 05 <46>
2308:05 05 05 05 05 05 05 05 <4E>
2310:05 05 17 17 17 17 17 17 <C8>
2318:17 17 17 17 17 17 17 17 <5E>
2320:17 17 17 17 05 08 05 17 <75>
2328:17 17 17 17 17 17 17 17 <4E>
2330:17 17 17 17 17 17 17 17 <F5>
2338:17 17 92 95 95 95 95 95 <98>
2340:95 95 92 92 92 92 92 92 <92>
2348:95 16 16 16 05 05 05 16 <3F>
2350:16 95 95 92 92 95 95 95 <37>
2358:92 92 95 95 95 92 95 92 <FA>
2360:95 92 95 95 95 95 92 92 <67>
2368:92 95 95 16 16 16 16 16 <4C>
2370:16 16 16 16 05 08 05 16 <D3>
2378:16 16 16 16 16 16 16 16 <BE>
2380:16 16 16 95 95 95 95 92 <1B>
2388:95 95 95 95 92 92 95 16 <84>
2390:16 16 16 16 16 16 16 16 <D6>
2398:16 16 16 16 05 05 05 16 <EF>
23A0:16 16 16 16 16 16 16 16 <E6>
23A8:16 16 16 16 16 95 92 92 <41>
23B0:92 95 92 92 95 95 95 16 <92>
23B8:16 01 04 00 16 16 16 16 <95>
23C0:16 16 16 01 04 04 04 00 <A1>
23C8:16 16 16 16 16 16 16 01 <D8>
23D0:04 00 16 16 16 95 95 95 <01>
23D8:95 95 95 92 92 92 16 16 <E6>
23E0:04 75 76 77 04 16 16 16 <42>
23E8:16 16 16 04 5A 5B 5C 04 <9C>
23F0:16 16 16 16 16 16 04 75 <70>
23F8:76 77 04 16 16 16 95 92 <BF>
2400:95 92 95 95 95 92 16 16 <48>
2408:04 78 4C 7A 04 04 04 04 <CD>
2410:04 04 04 04 5D 4C 5F 04 <F6>
2418:04 04 04 04 04 04 04 78 <D4>
2420:4C 7A 04 16 16 16 95 92 <52>
2428:95 92 95 95 95 95 16 16 <4C>
2430:04 7B 7C 7D 04 16 16 16 <0A>
2438:16 16 16 04 60 61 62 04 <61>
2440:16 16 16 16 16 16 04 7B <C9>
2448:7C 7D 04 16 16 16 95 92 <C4>
2450:92 95 92 92 92 95 16 16 <9B>
2458:16 02 04 03 16 16 16 16 <26>
2460:16 16 16 16 02 04 03 16 <99>
2468:16 16 16 16 16 16 16 02 <5C>
2470:04 03 16 16 16 16 92 92 <57>
2478:92 92 92 95 92 95 92 16 <DB>
2480:16 16 04 16 16 16 16 09 <75>
2488:0A 0B 16 16 16 04 16 16 <BF>
2490:16 09 0A 0B 16 16 16 16 <63>
2498:04 16 16 16 16 92 95 92 <42>
24A0:95 92 95 95 95 95 95 16 <AF>
24A8:16 16 04 16 16 16 16 0C <60>
24B0:4C 0E 16 16 16 04 16 16 <49>
24B8:16 0C 4C 0E 16 16 16 16 <C3>
24C0:04 16 16 16 92 92 92 92 <41>
24C8:95 95 95 95 92 92 92 16 <1C>
24D0:16 16 04 16 16 16 3C 0F <19>
24D8:10 11 16 16 16 04 16 16 <83>
24E0:16 0F 10 11 16 16 16 16 <55>
24E8:04 16 16 92 92 95 95 95 <36>
24F0:92 95 92 92 92 16 16 16 <00>
24F8:16 16 04 16 16 16 3C 16 <48>
2500:16 16 16 16 16 04 16 16 <01>
2508:16 16 16 16 16 16 16 16 <4E>
2510:04 16 16 16 92 92 95 92 <94>
2518:92 92 92 92 92 3E 3E 3E <2E>
2520:16 16 04 16 16 16 3C 16 <72>
2528:16 16 16 16 16 04 16 16 <E9>
2530:16 16 16 16 16 16 16 16 <F9>
2538:04 16 16 16 16 16 92 92 <EC>
2540:95 95 95 95 3E 3E 04 04 <85>
2548:04 04 04 04 04 04 04 04 <92>
2550:04 04 04 04 36 37 38 04 <59>
2558:04 04 04 04 04 04 04 04 <A2>
2560:04 04 04 04 16 16 16 95 <38>
2568:95 95 95 95 3E 3E 04 3E <D7>
2570:16 16 16 16 16 16 3C 3C <24>
2578:16 3C 16 16 39 3A 3B 16 <2E>
2580:16 16 16 16 16 16 16 16 <C2>
2588:16 16 16 04 16 16 16 95 <30>
2590:92 92 92 92 16 36 37 38 <A3>
2598:16 16 16 16 16 16 3C 6C <7D>
25A0:6D 6E 16 16 3C 3D 3E 16 <38>
25A8:16 6C 6D 6E 16 16 16 16 <98>
25B0:16 16 36 37 38 16 16 92 <1D>
25B8:92 92 92 95 16 39 4C 3B <E4>
25C0:16 16 16 16 3C 16 16 6F <8C>
25C8:70 71 16 16 16 04 16 16 <CD>
25D0:16 6F 70 71 16 16 16 16 <21>
25D8:16 16 39 4C 3B 16 16 92 <86>
25E0:95 95 95 95 16 3C 3D 3E <4E>
25E8:16 16 16 16 16 16 3C 72 <98>
25F0:73 74 16 16 16 04 16 16 <F7>
25F8:16 72 73 74 16 16 16 16 <E2>
2600:16 16 3C 3D 3E 16 16 92 <20>
2608:95 92 95 95 95 16 04 04 <14>
2610:04 04 04 04 04 04 04 16 <5E>
2618:16 3C 16 16 6C 6D 6E 16 <8C>
2620:16 16 16 16 04 04 04 04 <4D>
2628:04 04 04 04 16 16 16 92 <FF>
2630:92 92 92 92 92 16 16 16 <7C>
2638:16 16 16 16 16 16 04 16 <60>
2640:16 3C 16 16 6F 4C 71 16 <6E>
2648:16 16 16 16 04 16 16 16 <03>
2650:16 16 16 16 16 16 16 92 <08>
2658:92 95 92 95 92 16 16 16 <C5>
2660:16 16 16 16 16 3F 40 41 <C0>
2668:16 3C 16 16 72 73 74 16 <0E>
2670:16 16 16 3F 40 41 16 16 <4D>
2678:16 16 16 16 16 95 92 92 <33>
2680:95 95 95 95 95 92 16 16 <91>
2688:16 16 16 16 16 42 43 44 <0D>
2690:04 04 04 16 16 04 16 16 <A4>
2698:04 04 04 42 43 44 16 16 <F9>
26A0:16 16 16 16 16 95 95 92 <41>
26A8:95 92 95 95 95 95 95 16 <EB>
26B0:16 16 16 16 16 45 46 47 <BA>
26B8:16 16 04 04 04 04 04 04 <91>
26C0:04 16 16 45 46 47 16 16 <3A>
26C8:16 16 16 92 95 92 95 92 <32>
26D0:95 95 95 92 92 92 92 92 <65>
26D8:92 92 16 16 16 16 16 16 <41>
26E0:16 16 16 16 16 04 16 16 <D3>
26E8:16 16 16 16 16 16 16 16 <F3>
26F0:16 16 95 92 92 92 95 92 <32>
26F8:92 95 95 92 95 95 95 95 <E3>
2700:92 95 95 16 16 16 16 16 <AB>
2708:16 16 16 16 3F 40 41 16 <9E>
2710:16 16 16 16 16 16 16 16 <5A>
2718:16 95 95 92 92 95 95 92 <E4>
2720:95 92 95 95 92 92 95 95 <96>
2728:92 92 95 95 16 16 16 16 <F2>
2730:16 16 16 16 42 4C 44 16 <92>
2738:16 16 16 16 16 16 16 95 <01>
2740:95 95 92 95 95 92 95 95 <FB>
2748:95 95 92 95 92 92 92 95 <B5>
2750:95 95 92 95 95 95 92 92 <CE>
2758:16 16 16 16 45 46 47 16 <43>
2760:16 16 16 95 95 95 95 95 <FA>
2768:92 95 95 92 92 95 95 92 <97>
2770:92 95 95 92 92 95 95 92 <5E>
2778:92 95 95 92 92 95 95 92 <46>
2780:93 16 16 16 16 16 16 16 <EC>
2788:16 16 95 95 95 95 95 95 <1E>
2790:95 95 95 95 92 95 95 92 <DC>
2798:92 92 95 92 92 95 16 16 <F4>
27A0:16 95 95 92 95 95 92 95 <91>
27A8:95 92 93 16 16 16 16 16 <04>
27B0:95 95 95 95 95 95 92 92 <4E>
27B8:92 16 16 16 92 95 95 92 <9F>
27C0:92 95 95 95 95 16 09 0A <B9>
27C8:0B 16 95 92 95 95 92 92 <AD>
27D0:95 95 92 93 16 16 16 92 <40>
27D8:95 95 92 95 95 92 92 95 <3E>
27E0:16 09 0A 0B 16 92 95 95 <24>
27E8:92 95 95 92 95 16 0C 4C <D9>
27F0:0E 16 16 16 16 16 16 16 <38>
27F8:16 16 16 16 16 16 16 16 <42>
2800:16 16 16 16 16 16 16 16 <50>
2808:16 0C 4C 0E 16 95 92 92 <7C>
2810:95 92 92 92 92 16 0F 10 <85>
2818:11 16 95 95 95 92 92 95 <FD>
2820:95 92 92 92 95 16 16 95 <40>
2828:95 92 92 92 95 95 92 92 <86>
2830:16 0F 10 11 16 92 92 95 <11>
2838:95 95 95 92 92 92 16 16 <7D>
2840:16 92 95 95 92 95 92 95 <A7>
2848:95 92 95 95 92 16 16 95 <69>
2850:95 92 92 92 95 92 92 95 <D8>
2858:95 16 16 16 95 92 92 92 <BF>
2860:92 92 92 92 95 95 92 92 <6F>
2868:95 95 92 92 95 95 92 95 <88>
2870:95 95 95 95 95 16 16 95 <1A>
2878:95 95 92 95 95 92 92 95 <9B>
2880:95 92 92 92 95 95 92 92 <D1>
2888:95 92 95 92 92 92 92 95 <E8>
2890:95 92 95 95 92 92 95 92 <0F>
2898:92 95 92 95 92 16 16 95 <00>
28A0:95 92 92 92 95 92 92 95 <A6>
28A8:95 92 92 92 95 95 92 92 <09>
28B0:92 95 92 92 95 95 92 92 <BE>
28B8:92 95 95 92 95 95 95 92 <1C>
28C0:92 95 95 92 92 16 16 95 <28>
28C8:95 95 92 95 95 92 92 95 <4C>
28D0:95 92 92 95 95 92 92 95 <85>
28D8:95 92 95 92 95 92 92 95 <AE>
28E0:95 92 95 95 92 92 95 92 <E3>
28E8:92 92 51 52 52 52 53 92 <F5>
28F0:92 92 92 92 92 92 95 95 <B6>
28F8:95 95 92 95 95 92 92 95 <1C>
2900:95 92 95 92 92 92 95 95 <71>
2908:95 95 92 92 92 92 95 95 <C8>
2910:95 95 54 48 49 4A 56 95 <81>
2918:95 92 92 92 92 95 92 92 <FC>
2920:92 95 92 92 95 92 92 92 <81>
2928:92 92 95 92 95 92 95 51 <1E>
2930:52 52 52 53 95 92 95 16 <AF>
2938:16 16 16 4B 70 4D 56 92 <79>
2940:92 92 92 92 92 92 51 52 <87>
2948:52 52 53 95 95 92 92 92 <A0>
2950:95 92 92 95 95 92 95 54 <C7>
2958:48 49 4A 56 95 92 92 16 <F9>
2960:95 92 54 4E 4F 50 56 92 <37>
2968:92 92 95 92 95 92 54 48 <D3>
2970:49 4A 56 95 95 95 92 92 <AA>
2978:95 92 95 95 92 95 92 54 <ED>
2980:4B 4C 4D 16 16 16 16 16 <A1>
2988:92 95 57 58 58 58 59 95 <9F>
2990:95 95 95 16 16 16 16 4B <C6>
2998:4C 4D 56 92 95 95 92 95 <7C>
29A0:92 95 92 95 95 95 92 54 <FE>
29A8:4E 4F 50 56 95 95 95 16 <0E>
29B0:92 92 92 92 92 92 92 92 <DD>
29B8:92 92 92 16 95 95 54 4E <7D>
29C0:4F 50 56 95 95 95 92 92 <57>
29C8:92 92 95 95 95 92 95 57 <34>
29D0:58 58 58 59 95 95 95 16 <2F>
29D8:92 92 95 95 92 95 95 92 <53>
29E0:92 95 95 16 95 95 57 58 <B4>
29E8:58 58 59 95 95 95 95 92 <8E>
29F0:95 92 95 92 95 95 95 95 <A6>
29F8:92 95 95 95 95 95 95 16 <8E>
2A00:92 95 95 92 92 95 95 92 <B8>
2A08:92 95 95 16 95 92 95 95 <07>
2A10:95 95 92 92 95 95 95 95 <E6>
2A18:92 95 95 92 92 92 95 95 <97>
2A20:95 92 92 95 92 92 95 16 <E0>
2A28:16 16 16 16 16 16 16 16 <EB>
2A30:16 16 16 16 16 16 16 16 <84>
2A38:95 95 92 92 95 95 92 95 <C3>
2A40:92 92 95 95 92 95 95 92 <E8>
2A48:92 92 95 95 92 92 95 16 <58>
2A50:92 95 95 95 92 92 95 16 <10>
2A58:95 95 95 92 92 92 92 16 <60>
2A60:95 95 92 95 92 95 95 92 <E6>
2A68:92 92 92 92 95 51 52 52 <74>
2A70:52 53 95 95 92 95 95 16 <AC>
2A78:92 92 92 92 92 92 95 16 <99>
2A80:95 92 92 92 95 95 95 16 <35>
2A88:92 95 95 92 51 52 52 52 <59>
2A90:53 95 95 95 92 54 48 49 <12>
2A98:4A 56 95 95 92 95 95 16 <4B>
2AA0:92 92 95 95 92 51 52 16 <B8>
2AA8:52 53 92 92 92 92 95 16 <D9>
2AB0:92 92 95 95 54 48 49 4A <74>
2AB8:56 92 92 92 95 54 4B 70 <DF>
2AC0:4D 16 16 16 16 16 16 16 <6F>
2AC8:92 95 92 92 95 54 48 49 <C3>
2AD0:4A 56 92 92 92 92 95 16 <54>
2AD8:16 16 16 16 16 4B 70 4D <DC>
2AE0:56 92 95 95 95 54 4E 4F <AF>
2AE8:50 56 92 95 95 95 92 92 <23>
2AF0:92 92 95 92 92 54 4B 70 <51>
2AF8:4D 56 95 92 92 95 95 92 <B2>
2B00:92 92 95 95 54 4E 4F 50 <50>
2B08:56 95 95 95 95 57 58 58 <69>
2B10:58 59 95 92 92 95 95 95 <D5>
2B18:92 92 92 92 92 54 4E 4F <AC>
2B20:50 56 92 92 92 95 95 92 <A9>
2B28:95 95 92 95 57 58 58 58 <DC>
2B30:59 92 92 92 95 95 92 92 <97>
2B38:92 95 95 92 92 95 95 92 <8E>
2B40:92 92 95 92 92 57 58 58 <0A>
2B48:58 59 92 95 92 95 95 95 <6D>
2B50:95 92 95 95 92 92 92 95 <46>
2B58:95 92 95 95 95 92 92 95 <76>
2B60:95 92 95 92 95 95 92 95 <2A>
```

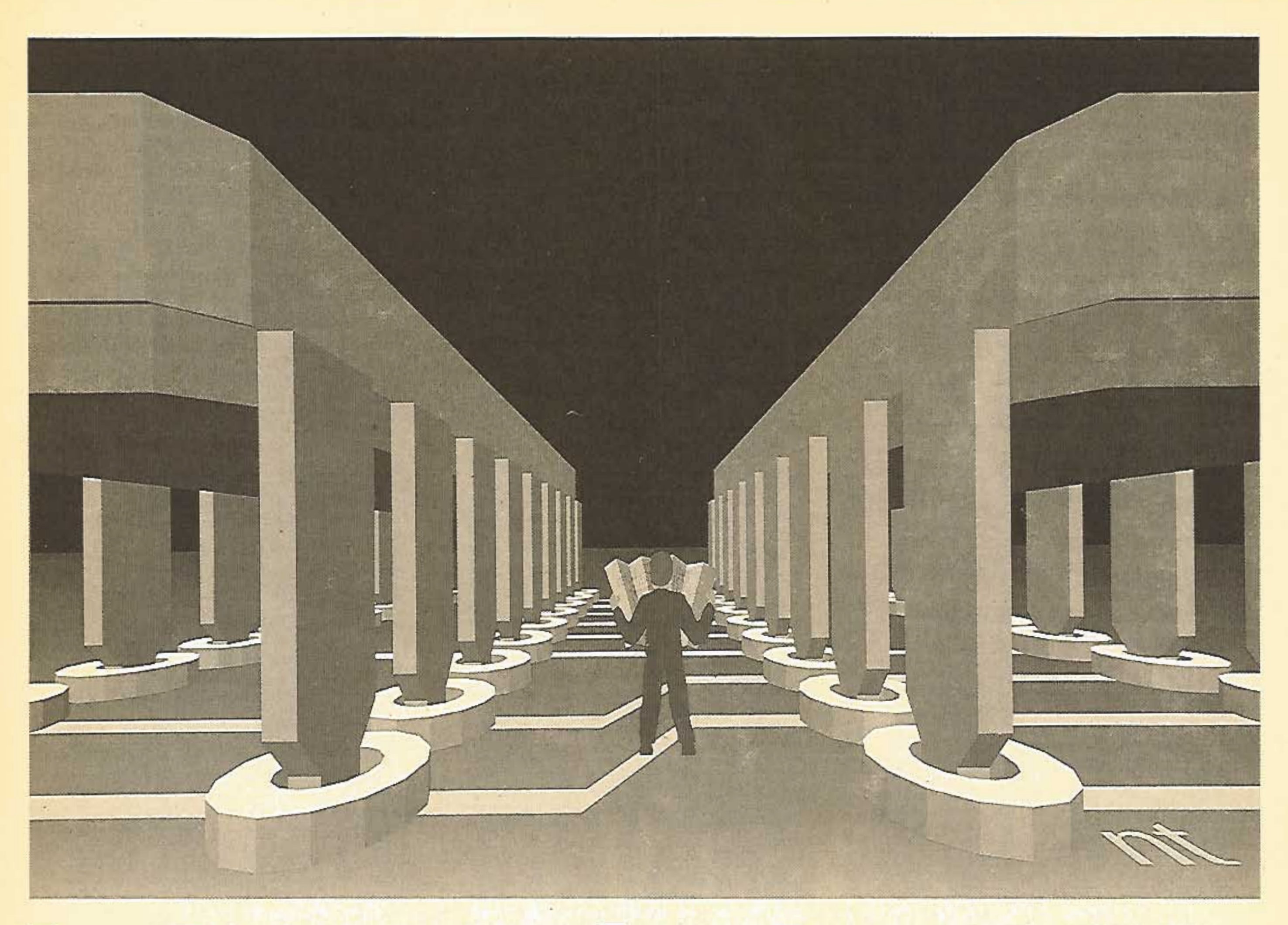
```
2B68:92 92 95 95 92 95 95 92 <CE>
2B70:92 95 92 92 92 95 95 92 <E6>
2B78:92 95 95 92 95 92 92 95 <57>
2B80:95 95 92 92 92 92 95 92 <49>
2B88:92 92 95 92 95 92 95 95 <86>
2B90:92 92 92 92 95 92 92 95 <18>
2B98:92 16 16 16 16 16 16 16 <9C>
2BA0:92 92 92 95 95 92 92 92 <75>
2BA8:95 92 92 95 95 95 92 95 <32>
2BB0:16 16 16 16 16 95 92 95 <75>
2BB8:92 95 92 92 95 92 92 16 <38>
2BC0:16 16 16 16 16 16 16 16 <15>
2BC8:16 95 92 92 95 95 92 92 <5F>
2BD0:16 16 16 92 92 95 16 16 <BB>
2BD8:16 16 16 16 16 16 16 16 <0D>
2BE0:95 95 95 95 92 92 16 16 <FE>
2BE8:16 16 16 16 16 16 16 16 <3C>
2BF0:16 16 16 16 16 16 16 16 <42>
2BF8:16 16 16 16 16 16 16 16 <2C>
2C00:16 16 16 16 16 16 16 16 <58>
2C08:16 16 16 16 16 16 16 16 <60>
2C10:16 16 16 16 16 16 16 16 <68>
2C18:16 16 16 16 16 16 16 16 <2E>
2C20:16 16 00 00 00 00 00 00 <88>
2C28:00 00 38 30 0D 0D 20 FE <65>
2C30:A0 20 14 A2 20 A1 A0 20 <70>
2C38:B0 9B A9 02 8D 07 D2 A9 <91>
2C40:08 8D 06 D2 20 09 A2 20 <56>
2C48:12 9B 20 5C 9F 20 CC 9F <CE>
2C50:20 A9 9C 20 B3 9E 20 42 <0E>
2C58:99 20 20 9A AD 64 9A D0 <DD>
2C60:06 20 87 9E 4C 16 99 20 <62>
2C68:64 99 4C 16 99 4C 3F 99 <EF>
2C70:AD FC 02 C9 1C F0 01 60 <ED>
2C78:A9 00 8D 1A 02 A9 FF 8D <BA>
2C80:FC 02 20 09 A2 AD FC 02 <A9>
2C88:C9 FF F0 F9 A9 02 8D 1A <1B>
2C90:02 60 AD 84 02 F0 15 A9 <62>
2C98:00 A8 AE F6 99 9D 00 B6 <F1>
2CA0:E8 C8 C0 20 D0 F7 A9 00 <8C>
2CA8:8D 01 D2 60 18 AD 9F A0 <95>
2CB0:69 32 8D 02 D0 18 AD F7 <0A>
2CB8:99 6D A0 A0 69 40 8D F6 <7B>
2CC0:99 AA A0 00 B9 F8 99 9D <B7>
2CC8:00 B6 E8 C8 C0 28 D0 F4 <39>
2CD0:18 AD F7 99 69 02 29 07 <6C>
2CD8:8D F7 99 D0 37 18 AD A0 <D8>
2CE0:A0 4A 4A 4A A8 A9 E7 79 <97>
2CE8:61 9E 85 CB A9 A3 79 74 <CA>
2CF0:9E 85 CC 18 AD 9F A0 69 <FF>
2CF8:04 4A 4A A8 B1 CB 29 7F <32>
2D00:AA BD 00 06 C9 0A F0 0C <19>
2D08:A9 1F 91 CB A9 05 8D 89 <4A>
2D10:9B 20 86 9D 18 A9 64 8D <01>
2D18:00 D2 AD F7 99 0A 05 80 <83>
2D20:8D 01 D2 60 B2 00 00 00 <E6>
2D28:00 00 24 24 00 00 24 24 <B5>
2D30:00 00 00 42 42 00 00 00 <B8>
2D38:42 42 00 00 00 00 81 81 <88>
2D40:00 00 00 00 81 81 00 00 <A8>
2D48:00 00 00 00 00 00 AD FC <F9>
2D50:02 C9 21 F0 06 A9 FF 8D <42>
2D58:FC 02 60 A9 00 8D 27 9F <E5>
2D60:8D 01 D2 A9 B6 85 CC 18 <95>
2D68:AD A0 A0 69 58 85 CB 20 <36>
2D70:1B 9F 18 AD 64 9A 49 01 <F6>
2D78:8D 64 9A 0A 0A AA A0 00 <3C>
2D80:BD 65 9A 99 7B A3 E8 C8 <15>
2D88:C0 04 D0 F4 A9 FF 8D FC <75>
2D90:02 60 00 22 2F 2D 22 00 <54>
2D98:27 35 2E A9 60 8D 17 A1 <A8>
2DA0:A9 00 8D 02 D0 8D 04 D0 <91>
2DA8:8D 05 D0 8D 06 D0 8D 07 <6E>
2DB0:D0 A9 8F 8D 01 D2 A9 00 <D9>
2DB8:8D 03 D2 8D 05 D2 8D 07 <3E>
2DC0:D2 18 AD A0 A0 69 58 AA <BC>
2DC8:A8 20 09 A2 AD 0A D2 8D <7F>
2DD0:00 D2 9D 00 B4 99 00 B4 <EE>
2DD8:AD 0A D2 9D 00 B5 99 00 <23>
2DE0:B5 AD 0A D2 9D 00 B7 99 <39>
2DE8:00 B7 E0 3D F0 01 CA C0 <D7>
2DF0:DE F0 04 C8 4C 9B 9A E0 <16>
2DF8:3D D0 CE 18 AD A0 A0 69 <C6>
2E00:58 AA A8 A9 00 20 09 A2 <66>
2E08:A9 00 9D 00 B4 99 00 B4 <00>
2E10:9D 00 B5 99 00 B5 9D 00 <F6>
2E18:B7 99 00 B7 8C 00 D2 E0 <8B>
2E20:3D F0 01 CA C0 DE F0 04 <AC>
2E28:C8 4C D7 9A E0 3D D0 D5 <DD>
2E30:A9 00 8D 01 D2 85 14 A5 <94>
2E38:14 C9 50 D0 FA 4C 00 A9 <7D>
2E40:A2 00 18 BD 08 D0 C9 04 <E3>
2E48:F0 04 C9 00 D0 06 E8 E0 <FD>
2E50:04 D0 EF 60 18 AD A0 A0 <D9>
2E58:69 08 DD 46 9D B0 19 A9 <EE>
2E60:00 9D 04 D0 9D 4A 9D 38 <E8>
2E68:AD D8 9B D0 03 4C 6D 9A <1F>
2E70:E9 05 8D D8 9B 20 B0 9B <09>
2E78:60 AD 89 9B F0 18 18 AD <77>
2E80:89 9B 0A 0A 0A 0A 49 FF <B3>
2E88:8D 02 D2 A9 8F 8D 03 D2 <FB>
2E90:CE 89 9B 4C 6D 9B A9 00 <6F>
2E98:8D 03 D2 AD 8A 9B F0 11 <B4>
2EA0:A9 DC 8D 04 D2 AD 8A 9B <2D>
2EA8:09 80 8D 05 D2 CE 8A 9B <20>
2EB0:60 A9 00 8D 05 D2 60 00 <6F>
2EB8:00 5B 18 AD 9F 9B 69 01 <F3>
2EC0:29 0F 8D 9F 9B A8 B9 A0 <88>
2EC8:9B 8D C3 02 60 0B 30 32 <47>
2ED0:34 36 38 3A 3C 3C 3C 3C <FD>
2ED8:3A 38 36 34 32 30 A9 00 <EC>
2EE0:A8 99 8E A3 C8 C0 1D D0 <9D>
2EE8:F8 A9 C2 A0 00 CC D8 9B <68>
2EF0:F0 07 99 8E A3 C8 4C BF <A5>
2EF8:9B EE D9 9B AC D9 9B B9 <9A>
2F00:DA 9B 8D F9 A2 60 19 01 <13>
2F08:00 92 92 92 B2 12 32 18 <84>
2F10:AC 41 9D B9 46 9D 69 3F <6A>
2F18:AA AC 41 9D BD 00 B3 39 <F6>
2F20:5E 9D 9D 00 B3 BD 01 B3 <25>
2F28:39 5E 9D 9D 01 B3 BD 02 <6A>
2F30:B3 39 5E 9D 9D 02 B3 A2 <30>
2F38:00 AC 41 9D AD 0A D2 29 <CD>
2F40:0F 8D A8 9C AD 0A D2 29 <7A>
2F48:03 6D A8 9C 99 46 9D A8 <D9>
2F50:18 A9 E7 79 61 9E 85 CD <B6>
2F58:A9 A3 79 74 9E 85 CE AC <98>
2F60:41 9D 18 AD 0A D2 29 1F <5F>
2F68:8D A8 9C AD 0A D2 29 03 <62>
2F70:6D A8 9C 99 42 9D A8 B1 <C1>
2F78:CD C9 4C F0 06 E8 E0 10 <93>
2F80:D0 B7 60 A9 07 8D 8A 9B <53>
2F88:AC 41 9D 18 B9 46 9D 0A <09>
2F90:0A 0A 99 46 9D B9 42 9D <D5>
2F98:0A 0A 99 42 9D 18 AD 9F <F3>
2FA0:A0 8D A7 9C B9 42 9D 20 <F0>
2FA8:97 9C 99 4E 9D 18 AD A0 <29>
2FB0:A0 69 10 8D A7 9C B9 46 <E2>
2FB8:9D 20 97 9C 99 52 9D A9 <91>
2FC0:01 99 4A 9D 60 CD A7 9C <F1>
2FC8:F0 05 90 06 A9 02 60 A9 <70>
2FD0:00 60 A9 01 60 56 05 A9 <5A>
2FD8:00 8D 41 9D AC 41 9D B9 <8F>
2FE0:4A 9D C9 01 F0 06 20 E1 <6C>
2FE8:9B 4C 0D 9D 18 AE 41 9D <A7>
2FF0:BC 4E 9D B9 57 9D 8D D4 <03>
2FF8:9C BC 52 9D B9 57 9D 8D <12>
3000:D7 9C FE 42 9D FE 46 9D <DB>
3008:AC 41 9D BE 5A 9D 18 B9 <F8>
3010:42 9D 69 30 99 04 D0 18 <1D>
3018:A9 B3 85 D0 B9 46 9D 69 <CA>
3020:3C 85 CF B9 5E 9D 8D 56 <37>
3028:9D A0 00 B1 CF 2D 56 9D <FD>
3030:1D 62 9D 91 CF E8 C8 C0 <BB>
3038:09 D0 F0 AC 41 9D B9 42 <FB>
3040:9D C9 02 90 12 C9 98 B0 <7F>
3048:0E B9 46 9D C9 02 90 07 <CE>
3050:C9 92 B0 03 4C 2E 9D A9 <D5>
3058:00 99 4A 9D EE 41 9D AC <2D>
3060:41 9D C0 04 F0 03 4C AE <92>
3068:9C A9 01 8D 1E D0 60 04 <AC>
3070:8F 2F 37 43 1F 1B 23 2F <D5>
3078:01 01 01 01 01 01 01 01 <D8>
3080:01 01 01 01 3F BD FE DE <98>
3088:00 09 12 1B FC F3 CF 3F <A7>
3090:00 00 00 03 03 03 00 00 <45>
3098:00 00 00 00 0C 0C 0C 00 <A1>
30A0:00 00 00 00 00 30 30 30 <51>
30A8:00 00 00 00 00 00 C0 C0 <49>
30B0:C0 00 00 00 18 AC 64 9A <44>
30B8:F8 AD D5 9D 79 D7 9D 8D <D1>
30C0:D5 9D AD D6 9D 69 00 8D <FC>
30C8:D6 9D D8 AD D6 9D 20 C2 <6B>
30D0:9D 8C 65 A3 8C EF AB 8E <BC>
30D8:66 A3 8E F0 AB AD D5 9D <13>
30E0:20 C2 9D 8C 67 A3 8C F1 <07>
30E8:AB 8E 68 A3 8E F2 AB 60 <2B>
30F0:18 AA 29 F0 4A 4A 4A 4A <44>
30F8:69 10 A8 18 8A 29 0F 69 <7D>
3100:10 AA 60 30 00 0A 01 18 <05>
3108:AD 28 9F 69 05 4A 4A A8 <D3>
3110:8C 45 9E 18 AD 29 9F 4A <31>
3118:4A 4A AA A9 E7 7D 61 9E <FF>
3120:85 CB A9 A3 7D 74 9E 85 <E9>
3128:CC B1 CB 29 7F A8 B9 00 <19>
3130:06 C9 0A D0 01 60 AC 45 <2B>
3138:9E AA A9 0F 8D 89 9B 38 <56>
3140:A5 CB E9 52 85 CB A5 CC <D7>
3148:E9 00 85 CC 18 A5 CB 7D <D6>
3150:46 9E 85 CB A5 CC 69 00 <E9>
3158:85 CC A2 00 18 BD 58 9E <A7>
3160:6D 45 9E A8 8A 69 1B 91 <9A>
3168:CB E8 E0 09 D0 EE 20 86 <B3>
3170:9D 60 16 23 52 51 50 2A <C6>
3178:29 28 02 01 00 00 00 00 <C9>
3180:00 01 01 01 01 01 00 01 <60>
3188:02 28 29 2A 50 51 52 28 <48>
3190:50 78 A0 C8 F0 18 40 68 <F1>
3198:90 B8 E0 08 30 58 80 A8 <F5>
31A0:D0 F8 00 00 00 00 00 00 <68>
31A8:01 01 01 01 01 01 02 02 <0D>
31B0:02 02 02 02 02 AD 84 02 <C5>
31B8:F0 01 60 AD 27 9F C9 00 <6E>
31C0:F0 01 60 A9 10 8D 27 9F <74>
31C8:18 AD 9F A0 69 32 8D 02 <86>
31D0:D0 AD 9F A0 8D 28 9F 18 <93>
31D8:AD A0 A0 69 58 8D 29 9F <19>
31E0:60 AD 27 9F D0 01 60 18 <B7>
31E8:A9 B6 85 CC AD 29 9F 85 <4D>
31F0:CB 20 1B 9F CE 29 9F CE <19>
31F8:29 9F AD 27 9F A8 09 80 <C4>
3200:8D 01 D2 98 0A 0A 8D 26 <66>
3208:9F 38 A9 64 ED 26 9F 8D <BF>
3210:00 D2 BE 2A 9F A0 00 BD <77>
3218:3C 9F 91 CB C8 E8 C0 08 <0E>
3220:D0 F5 CE 27 9F AD 27 9F <E9>
3228:C9 01 D0 1C 20 1B 9F 38 <50>
3230:AD 29 9F E9 3C 8D 29 9F <1D>
3238:20 D9 9D A9 00 8D 27 9F <44>
3240:A9 00 8D 02 D0 8D 01 D2 <B3>
3248:60 A9 00 A8 91 CB C8 C0 <88>
3250:10 D0 F9 60 08 00 89 44 <5D>
3258:18 18 18 10 10 10 08 08 <AB>
3260:08 08 08 00 00 00 00 00 <CB>
3268:00 95 18 3C 7E FF FF 7E <64>
3270:3C 18 00 18 3C 7E 7E 3C <8F>
3278:18 00 00 00 18 3C 3C 18 <2B>
3280:00 00 00 00 00 18 18 00 <75>
3288:00 00 AD 78 02 C9 0F D0 <2B>
3290:01 60 A8 18 AD 9F A0 79 <5F>
3298:AC 9F 8D 9F A0 18 AD A0 <8C>
32A0:A0 79 BC 9F 8D A0 A0 AD <E7>
32A8:9F A0 C9 02 90 10 C9 90 <FB>
32B0:B0 14 AD A0 A0 C9 02 90 <BD>
32B8:15 C9 6C B0 17 60 A9 02 <CE>
32C0:8D 9F A0 4C 84 9F A9 90 <CB>
32C8:8D 9F A0 4C 84 9F A9 02 <35>
32D0:8D A0 A0 60 A9 6C 8D A0 <AD>
32D8:A0 60 00 00 00 00 00 01 <25>
32E0:01 01 00 FF FF FF 00 00 <E8>
32E8:00 00 00 00 00 00 00 02 <4E>
32F0:FE 00 00 02 FE 00 00 02 <65>
32F8:FE 00 18 AD 9F A0 69 2E <A5>
3300:8D 00 D0 8D 03 D0 69 08 <B1>
3308:8D 01 D0 AD A0 A0 69 3C <4A>
3310:A8 A2 00 BD FE 9F 99 00 <6A>
3318:B4 BD 35 A0 99 00 B5 BD <1B>
3320:6A A0 99 00 B7 C8 E8 E0 <5F>
3328:35 D0 E8 60 00 00 00 06 <16>
3330:0D 09 06 06 09 0D 06 00 <E8>
3338:00 00 00 00 00 00 00 00 <9E>
3340:00 00 00 00 00 00 00 00 <A6>
3348:00 00 00 00 00 00 00 01 <AF>
3350:01 02 06 04 02 05 0B 16 <08>
3358:16 34 44 5E 32 01 00 00 <CA>
3360:00 00 00 00 00 00 60 B0 <37>
3368:90 60 60 90 B0 60 00 00 <B2>
3370:00 00 00 00 00 00 00 00 <D6>
3378:00 00 00 00 00 00 00 00 <DE>
3380:00 00 00 00 00 00 80 80 <67>
3388:40 60 20 40 A0 D0 68 68 <23>
3390:2C 22 3A 4C 80 00 00 00 <15>
3398:00 00 00 00 00 00 00 00 <FE>
33A0:00 00 00 00 00 00 00 00 <06>
33A8:00 00 00 00 00 00 00 00 <0E>
33B0:00 00 00 00 00 00 00 00 <16>
33B8:00 00 00 00 18 18 18 18 <87>
33C0:18 3C 3C 3C 3C 3C 7E 7E <98>
33C8:18 00 00 00 00 83 46 A9 <79>
33D0:3E 8D 2F 02 A9 01 8D 6F <92>
33D8:02 A9 03 8D 1D D0 A9 B0 <BC>
33E0:8D 07 D4 A9 0A 8D C0 02 <57>
33E8:8D C1 02 A9 36 8D C3 02 <07>
33F0:A9 1A 8D C2 02 A9 64 8D <F5>
33F8:9F A0 A9 00 8D A0 A0 A9 <7D>
3400:01 8D 0B D0 A9 55 8D 0C <2E>
3408:D0 A0 00 98 99 00 B3 99 <96>
3410:00 B4 99 00 B5 99 00 B6 <5E>
3418:99 00 B7 C8 D0 EE A9 64 <94>
3420:8D A0 A0 A9 48 8D 9F A0 <45>
3428:20 CC 9F 60 A0 00 A9 16 <D3>
3430:99 E7 A3 99 AF A4 99 77 <B9>
3438:A5 99 3F A6 99 8F A6 C8 <E2>
3440:C0 C8 D0 EC 60 EA 18 AD <FB>
3448:5D A1 69 01 29 07 8D 5D <A2>
3450:A1 F0 06 49 07 8D 05 D4 <A2>
3458:60 38 AD 07 A2 E9 28 8D <91>
```

»Arax« (Fortsetzung)

```
3460:07 A2 AD 08 A2 E9 00 8D <4C>
3468:08 A2 20 5E A1 A9 07 8D <8E>
3470:05 D4 AD 08 A2 C9 6C D0 <3F>
3478:11 AD 07 A2 C9 E4 D0 0A <1F>
3480:A9 F4 8D 07 A2 A9 98 8D <AC>
3488:08 A2 60 06 AD 07 A2 85 <FB>
3490:CB AD 08 A2 85 CC A0 27 <62>
3498:B1 CB 99 BF A3 88 10 F8 <FC>
34A0:A0 00 B9 07 A7 99 2F A7 <80>
34A8:B9 DF A6 99 07 A7 B9 B7 <FE>
34B0:A6 99 DF A6 B9 8F A6 99 <B0>
34B8:B7 A6 B9 67 A6 99 8F A6 <5B>
34C0:B9 3F A6 99 67 A6 B9 17 <EE>
34C8:A6 99 3F A6 B9 EF A5 99 <1F>
34D0:17 A6 B9 C7 A5 99 EF A5 <03>
34D8:B9 9F A5 99 C7 A5 B9 77 <37>
34E0:A5 99 9F A5 B9 4F A5 99 <F6>
34E8:77 A5 B9 27 A5 99 4F A5 <49>
34F0:B9 FF A4 99 27 A5 B9 D7 <88>
34F8:A4 99 FF A4 B9 AF A4 99 <D7>
3500:D7 A4 B9 87 A4 99 AF A4 <11>
3508:B9 5F A4 99 87 A4 B9 37 <33>
3510:A4 99 5F A4 B9 0F A4 99 <C3>
3518:37 A4 B9 E7 A3 99 0F A4 <E6>
3520:B9 BF A3 99 E7 A3 B9 97 <C3>
3528:A3 99 BF A3 C8 C0 28 F0 <18>
3530:03 4C 74 A1 60 D4 95 A9 <E6>
3538:00 85 14 A5 14 C9 01 D0 <26>
3540:FA 60 A9 04 8D 30 02 A9 <E7>
3548:A3 8D 31 02 A9 32 8D C6 <9A>
3550:02 A9 AF 8D 00 02 A9 A2 <A1>
3558:8D 01 02 A9 EA 8D 17 A1 <64>
3560:A9 00 8D EF A2 8D 27 9F <E0>
3568:8D D5 9D 8D D6 9D 8D 64 <5C>
3570:9A 8D 89 9B 8D 8A 9B 8D <F4>
3578:D9 9B 8D F7 99 20 4A 9A <A7>
3580:A9 92 8D F9 A2 A9 19 8D <53>
3588:D8 9B A9 00 A8 99 42 9D <83>
3590:C8 C0 10 D0 F8 A9 C0 8D <4E>
3598:0E D4 A9 F4 8D 07 A2 A9 <07>
35A0:98 8D 08 A2 A9 00 8D 5D <92>
35A8:A1 A9 8B 8D 28 02 A9 A2 <7D>
35B0:8D 29 02 A9 01 8D 1A 02 <66>
35B8:60 8D AC A2 8E AD A2 8C <35>
35C0:AE A2 20 17 A1 20 4B 9B <D6>
35C8:20 8C 9B A9 03 8D 1A 02 <E3>
35D0:AD AC A2 AE AD A2 AC AE <C0>
35D8:A2 60 00 02 00 8D ED A2 <F9>
35E0:8C EE A2 AC EF A2 AD 0B <FE>
35E8:D4 8D 0A D4 B9 F0 A2 8D <23>
35F0:16 D0 B9 F4 A2 8D 17 D0 <01>
35F8:B9 F8 A2 8D 18 D0 B9 FC <37>
3600:A2 8D 19 D0 B9 00 A3 8D <22>
3608:09 D4 C8 C0 04 D0 02 A0 <E6>
3610:00 8C EF A2 AD ED A2 AC <61>
3618:EE A2 40 00 04 00 00 00 <8C>
3620:B6 00 0C 0A 08 00 32 92 <40>
3628:04 00 00 00 B2 00 E0 E0 <C5>
3630:B8 00 70 70 70 80 4C AB <AD>
3638:A3 C2 5B A3 4B AB A3 42 <18>
3640:83 A3 CC AB A3 70 64 E7 <D1>
3648:A3 24 24 24 24 24 24 24 <F3>
3650:24 24 24 24 24 24 24 24 <B8>
3658:24 24 24 24 04 80 41 04 <2B>
3660:A3 00 00 52 52 52 52 52 <05>
3668:52 52 52 52 52 52 52 52 <82>
3670:52 52 52 52 52 52 52 52 <37>
3678:52 52 52 52 52 52 52 52 <52>
3680:52 52 52 52 52 52 52 00 <18>
3688:00 00 00 00 00 33 23 2F <35>
3690:32 25 1A 10 10 13 10 10 <FF>
3698:10 00 00 00 00 00 00 00 <0C>
36A0:00 37 25 21 30 2F 2E 1A <35>
36A8:00 22 2F 2D 22 00 00 00 <46>
36B0:00 00 00 00 00 33 28 29 <62>
36B8:25 2C 24 1A C2 C2 C2 C2 <02>
36C0:C2 C2 C2 C2 C2 C2 C2 C2 <70>
36C8:C2 C2 C2 C2 C2 C2 C2 C2 <58>
36D0:C2 C2 C2 C2 C2 00 00 00 <F7>
36D8:00 00 00 00 00 00 00 00 <44>
36E0:00 00 00 00 00 00 00 00 <4C>
36E8:00 00 00 00 00 C2 C2 C2 <A4>
36F0:C2 C2 C2 C2 C2 C2 C2 C2 <60>
36F8:C2 C2 C2 C2 C2 00 00 00 <58>
3700:00 00 00 00 00 00 00 00 <6E>
3708:00 00 00 00 00 00 00 00 <76>
3710:00 00 00 00 00 95 92 92 <87>
3718:92 95 95 95 95 92 92 92 <BB>
3720:92 95 95 95 95 54 48 49 <F0>
3728:4A 56 95 95 92 92 92 92 <88>
3730:95 92 92 92 95 92 92 95 <D6>
3738:92 92 92 92 92 95 92 95 <4C>
3740:92 95 51 52 52 52 53 95 <5F>
3748:92 95 16 16 16 16 4B 70 <58>
3750:4D 56 92 92 92 92 92 92 <CE>
3758:92 51 52 52 52 53 95 95 <E1>
3760:92 92 92 95 92 92 95 95 <4E>
3768:92 95 54 48 49 4A 56 95 <78>
3770:92 92 16 95 92 54 4E 4F <E7>
3778:50 56 92 92 92 95 92 95 <3C>
3780:92 54 48 49 4A 56 95 95 <93>
3788:95 92 92 95 92 95 95 92 <25>
3790:95 92 54 4B 4C 4D 16 16 <02>
3798:16 16 16 92 95 57 58 58 <81>
37A0:58 59 95 95 95 95 16 16 <79>
37A8:16 16 4B 4C 4D 56 92 95 <4E>
37B0:95 92 95 92 95 92 95 95 <CC>
37B8:95 92 54 4E 4F 50 56 95 <AE>
37C0:95 95 16 92 92 92 92 92 <7B>
37C8:92 92 92 92 92 92 16 95 <F3>
37D0:95 54 4E 4F 50 56 95 95 <D4>
37D8:95 92 92 92 92 95 95 95 <29>
37E0:92 95 57 58 58 58 59 95 <34>
37E8:95 95 16 92 92 95 95 92 <45>
37F0:95 95 92 92 95 95 16 95 <E9>
37F8:95 57 58 58 58 59 95 95 <B5>
3800:95 95 92 95 92 95 92 95 <0C>
3808:95 95 95 92 95 95 95 95 <AD>
3810:95 95 16 92 95 95 92 92 <4C>
3818:95 95 92 92 95 95 16 95 <C0>
3820:92 95 95 95 95 92 92 95 <D8>
3828:95 95 95 92 95 95 92 92 <04>
3830:92 95 95 95 92 92 95 92 <90>
3838:92 95 16 16 16 16 16 16 <82>
3840:16 16 16 16 16 16 16 16 <70>
3848:16 16 16 95 95 92 92 95 <FA>
3850:95 92 95 92 92 95 95 92 <0F>
3858:95 95 92 92 92 95 95 92 <63>
3860:92 95 16 92 95 95 95 92 <84>
3868:92 95 16 95 95 95 92 92 <86>
3870:92 92 16 95 95 92 95 92 <D8>
3878:95 95 92 92 92 92 92 95 <34>
3880:51 52 52 52 53 95 95 92 <C7>
3888:95 95 16 92 92 92 92 92 <10>
3890:92 95 16 95 92 92 92 95 <5D>
3898:95 95 16 92 95 95 92 51 <B3>
38A0:52 52 52 53 95 95 95 92 <C1>
38A8:54 1B 1C 1D 56 95 95 92 <1A>
38B0:95 95 16 92 92 95 95 92 <9A>
38B8:51 52 16 52 53 92 92 92 <E5>
38C0:92 95 16 92 92 95 95 54 <CE>
38C8:1B 1C 1D 56 92 92 92 95 <62>
38D0:54 1E 1F 20 16 16 16 16 <1B>
38D8:16 16 16 92 95 92 92 95 <5B>
38E0:54 1B 1C 1D 56 92 92 92 <00>
38E8:92 95 16 16 16 16 16 16 <B5>
38F0:1E 1F 20 56 92 95 95 95 <39>
38F8:54 21 22 23 56 92 95 95 <77>
3900:95 92 92 92 92 95 92 92 <34>
3908:54 1E 1F 20 56 95 92 92 <41>
3910:95 95 92 92 92 95 95 54 <6A>
3918:21 22 23 56 95 95 95 95 <24>
3920:57 58 58 58 59 95 92 92 <66>
3928:95 95 95 92 92 92 92 92 <DE>
3930:54 21 22 23 56 92 92 92 <3B>
3938:95 95 92 95 95 92 95 57 <81>
3940:58 58 58 59 92 92 92 95 <12>
3948:95 92 92 92 95 95 92 92 <C3>
3950:95 95 92 92 92 95 92 92 <63>
3958:57 58 58 58 59 92 95 92 <A9>
3960:95 95 95 95 92 95 95 92 <E8>
3968:92 92 95 95 92 95 95 95 <A5>
3970:92 92 95 95 92 95 92 95 <97>
3978:95 92 95 92 92 95 95 92 <E1>
3980:95 95 92 92 95 92 92 92 <7F>
3988:95 95 92 92 95 95 92 95 <C5>
3990:92 92 95 95 95 92 92 92 <81>
3998:92 95 92 92 92 95 92 95 <9F>
39A0:92 95 95 92 92 92 92 95 <EB>
39A8:92 92 95 92 16 16 16 16 <DE>
39B0:16 16 16 92 92 92 95 95 <2F>
39B8:92 92 92 95 92 92 95 95 <EE>
39C0:95 92 95 16 16 16 16 16 <C0>
39C8:95 92 95 92 95 92 92 95 <9C>
39D0:92 92 16 16 16 16 16 16 <1F>
39D8:16 16 16 16 95 92 92 95 <A2>
39E0:95 92 92 16 16 16 92 92 <72>
39E8:95 16 16 16 16 16 16 16 <99>
39F0:16 16 16 95 95 95 95 92 <C3>
39F8:92 16 16 16 16 16 16 16 <26>
3A00:16 16 16 16 16 16 16 16 <32>
3A08:16 16 16 16 16 16 16 16 <EB>
3A10:16 16 16 16 16 16 16 16 <84>
3A18:16 16 16 16 16 16 16 16 <7C>
3A20:16 16 16 16 16 16 16 16 <94>
3A28:16 16 16 16 16 16 16 16 <8C>
3A30:16 16 16 16 16 16 16 16 <A4>
3A38:16 16 16 16 16 16 16 16 <9C>
3A40:16 16 16 16 16 16 16 16 <74>
3A48:16 16 16 16 16 16 16 16 <BC>
3A50:16 16 16 16 16 16 16 16 <C4>
3A58:16 16 16 16 16 16 16 16 <AC>
3A60:16 16 16 16 16 16 16 16 <D4>
3A68:16 16 16 16 16 16 16 16 <DC>
3A70:16 16 16 16 16 16 16 16 <E4>
3A78:16 16 16 16 16 16 16 16 <CC>
3A80:16 16 16 16 16 92 95 92 <1D>
3A88:95 92 95 95 95 95 FF FF <DF>
3A90:00 A9 1F B0 20 A4 A9 AD <EB>
3A98:0B D4 8D 0A D4 C9 3C 90 <AB>
3AA0:23 E9 3C C9 1E B0 05 65 <F6>
3AA8:14 4C 1A A9 E5 14 A8 29 <F8>
3AB0:0F AA 98 4A 4A 4A 4A A8 <61>
3AB8:B9 84 A9 1D 94 A9 8D 1A <1F>
3AC0:D0 4C 03 A9 A9 00 8D 1A <9A>
3AC8:D0 AD 1F D0 C9 06 F0 08 <7E>
3AD0:AD 84 02 F0 03 4C 03 A9 <EF>
3AD8:A9 FF 8D 0E D4 A9 00 8D <C1>
3AE0:2F 02 85 14 A5 14 C9 05 <6A>
3AE8:D0 FA A9 0C 8D C4 02 A9 <E1>
3AF0:E0 8D F4 02 A9 0C 8D 30 <14>
3AF8:02 A9 AC 8D 31 02 A9 22 <DA>
3B00:8D 2F 02 A9 00 85 14 A5 <F2>
3B08:14 C9 64 D0 FA A9 00 8D <E3>
3B10:C8 02 4C 00 99 4C 81 A9 <96>
3B18:00 10 20 30 40 50 60 70 <09>
3B20:80 90 A0 B0 C0 D0 E0 F0 <DE>
3B28:00 01 02 03 04 05 06 07 <96>
3B30:07 06 05 04 03 02 01 00 <AE>
3B38:A9 00 8D 2F 02 8D 1D D0 <F2>
3B40:A9 23 8D 0F D2 A9 28 8D <68>
3B48:08 D2 A9 FF 8D 0E D4 A9 <69>
3B50:91 8D 30 02 A9 AA 8D 31 <D8>
3B58:02 A9 B8 8D F4 02 A0 00 <F0>
3B60:B9 8C AA 99 C4 02 C8 C0 <4A>
3B68:05 D0 F5 A9 15 8D 00 02 <6B>
3B70:A9 AA 8D 01 02 A9 43 8D <2D>
3B78:95 AA A9 AB 8D 96 AA A9 <44>
3B80:00 8D 85 AA 8D 81 AA 8D <90>
3B88:82 AA A9 C0 8D 0E D4 A9 <50>
3B90:E0 8D F4 02 A9 00 85 14 <A2>
3B98:A5 14 C9 02 D0 FA A9 00 <F6>
3BA0:8D 85 AA A9 22 8D 2F 02 <B1>
3BA8:60 8D 83 AA 8C 84 AA AC <1A>
3BB0:85 AA AD 0B D4 8D 0A D4 <56>
3BB8:B9 86 AA 8D 09 D4 B9 89 <CC>
3BC0:AA 8D 16 D0 C8 C0 03 D0 <55>
3BC8:02 A0 00 8C 85 AA 20 44 <42>
3BD0:AA AD 83 AA AC 84 AA 40 <58>
3BD8:38 AD 81 AA E9 01 29 07 <EC>
3BE0:8D 81 AA 8D 04 D4 C9 07 <13>
3BE8:D0 2A 18 AD 95 AA 69 01 <CB>
3BF0:8D 95 AA AD 96 AA 69 00 <FF>
3BF8:8D 96 AA EE 82 AA AD 82 <BC>
3C00:AA C9 B5 D0 0F A9 00 8D <FB>
3C08:82 AA A9 43 8D 95 AA A9 <6A>
3C10:AB 8D 96 AA 60 04 29 00 <5C>
3C18:0F 00 E0 B8 E0 0D 16 0D <0B>
3C20:16 08 04 00 02 70 80 70 <66>
3C28:56 6C AB 80 70 44 70 AE <73>
3C30:04 04 04 04 04 04 04 04 <A8>
3C38:84 70 46 B7 AA 70 07 70 <90>
3C40:06 70 06 70 06 70 06 06 <5F>
3C48:41 91 AA 00 00 00 23 2F <50>
3C50:30 39 32 29 27 28 34 00 <48>
3C58:11 19 18 17 00 00 00 00 <F3>
3C60:00 23 2F 2D 0D 21 32 34 <DF>
3C68:00 33 2F 26 34 37 21 32 <E4>
3C70:25 00 00 00 00 00 00 00 <7B>
3C78:37 32 29 34 34 25 2E 00 <83>
3C80:22 39 00 00 00 00 00 00 <D7>
3C88:00 00 00 2D 21 32 34 29 <32>
3C90:2E 00 28 2F 32 33 34 00 <B5>
3C98:00 00 00 00 00 00 00 00 <10>
3CA0:27 32 21 30 28 29 23 33 <B6>
3CA8:00 22 39 00 00 00 00 00 <CF>
3CB0:00 00 00 29 2E 27 2F 00 <27>
3CB8:33 23 28 2D 29 24 34 00 <A8>
3CC0:00 00 00 00 00 21 2E 24 <3D>
3CC8:00 2D 21 32 34 29 2E 00 <65>
3CD0:28 2F 32 33 34 00 00 23 <56>
3CD8:2F 2D 0D 21 32 34 00 33 <74>
3CE0:2F 26 34 37 21 32 25 00 <7F>
3CE8:30 32 25 33 25 2E 34 33 <19>
3CF0:0E 0E 0E 00 00 00 0D 00 <CE>
3CF8:21 32 21 38 00 0D 00 00 <E8>
3D00:00 37 32 29 34 34 25 2E <BB>
3D08:00 10 14 0E 18 17 00 22 <29>
3D10:39 00 2D 21 32 34 29 2E <B9>
3D18:00 28 2F 32 33 34 0E 00 <0A>
3D20:00 00 23 28 21 32 21 23 <A8>
3D28:34 25 32 24 25 33 29 27 <B9>
3D30:2E 00 22 39 00 29 2E 27 <C1>
3D38:2F 00 33 23 28 2D 29 24 <4F>
3D40:34 0E 00 00 2C 21 2E 24 <BE>
3D48:27 32 21 30 28 29 23 00 <34>
3D50:22 39 00 29 2E 27 2F 00 <29>
3D58:33 23 28 2D 29 24 34 00 <4F>
```

```
3D60:21 2E 24 00 2D 21 32 34 <01>
3D68:29 2E 00 28 2F 32 33 34 <21>
3D70:0E 00 00 00 00 00 2C 21 <69>
3D78:33 34 00 33 23 2F 32 25 <AB>
3D80:00 1A 00 10 10 13 10 10 <3F>
3D88:10 00 00 00 23 2F 2D 0D <45>
3D90:21 32 34 00 33 2F 26 34 <80>
3D98:37 21 32 25 00 30 32 25 <D8>
3DA0:70 70 70 70 70 70 70 70 <05>
3DA8:47 1A AC 41 0C AC 00 27 <0B>
3DB0:25 34 00 32 25 21 24 39 <DB>
3DB8:00 30 2C 21 39 25 32 00 <74>
3DC0:01 00 00 00 16 16 16 16 <05>
3DC8:16 16 16 16 16 16 16 16 <40>
3DD0:16 16 16 16 16 16 16 16 <46>
3DD8:16 16 16 16 16 16 16 16 <4E>
3DE0:16 92 92 92 92 92 16 09 <3D>
3DE8:0A 0B 16 16 16 16 16 16 <99>
3DF0:16 16 48 49 4A 16 16 16 <65>
3DF8:16 16 48 49 4A 16 16 16 <3D>
3E00:16 16 16 16 16 09 0A 0B <94>
3E08:16 16 95 92 95 92 16 0C <EF>
3E10:4C 0E 16 16 92 95 92 92 <F6>
3E18:16 16 4B 4C 4D 16 16 16 <48>
3E20:16 16 4B 4C 4D 16 16 92 <DC>
3E28:95 92 95 92 16 0C 4C 0E <0F>
3E30:16 16 92 92 92 95 16 0F <AE>
3E38:10 11 16 92 92 95 95 92 <4A>
3E40:16 16 4E 4F 50 16 16 16 <29>
3E48:16 16 4E 4F 50 16 16 92 <9D>
3E50:92 95 95 92 16 0F 10 11 <C4>
3E58:16 16 95 92 95 92 16 16 <66>
3E60:16 16 16 92 95 92 92 95 <06>
3E68:92 16 16 16 16 16 16 16 <9A>
3E70:16 16 16 16 16 16 95 92 <47>
3E78:95 92 95 92 16 16 16 16 <C2>
3E80:16 92 92 92 92 95 92 16 <B5>
3E88:16 16 16 92 95 92 95 92 <22>
3E90:92 16 16 16 16 16 16 16 <46>
3E98:16 16 16 16 16 16 92 92 <86>
3EA0:95 95 92 16 16 16 16 16 <02>
3EA8:16 95 95 92 92 95 92 92 <A0>
3EB0:16 16 16 16 16 16 16 16 <1A>
3EB8:16 16 16 16 16 16 16 16 <F3>
3EC0:16 16 16 16 16 16 16 16 <2A>
3EC8:16 16 16 16 16 16 16 16 <44>
3ED0:92 95 95 92 92 92 95 92 <B4>
3ED8:95 16 16 16 48 49 4A 16 <58>
3EE0:16 16 51 52 52 52 52 52 <0A>
3EE8:52 52 52 52 53 16 16 16 <34>
3EF0:48 49 4A 16 16 16 16 16 <58>
3EF8:95 92 92 92 92 95 95 92 <38>
3F00:16 16 16 16 4B 4C 4D 16 <EE>
3F08:16 16 54 51 52 52 52 52 <61>
3F10:52 52 52 53 56 16 16 16 <D8>
3F18:4B 4C 4D 16 16 16 16 16 <65>
3F20:92 95 95 92 92 95 95 16 <02>
3F28:16 16 16 16 4E 4F 50 16 <C1>
3F30:16 16 54 54 51 52 52 52 <B1>
3F38:52 52 53 56 56 16 16 16 <81>
3F40:4E 4F 50 16 16 16 16 16 <50>
3F48:16 92 95 92 92 92 16 16 <3F>
3F50:16 16 16 16 16 16 16 16 <AE>
3F58:16 16 54 54 54 51 52 52 <0D>
3F60:52 53 56 56 56 16 16 16 <B9>
3F68:16 16 16 16 16 16 16 16 <E6>
3F70:16 16 95 92 95 95 09 0A <56>
3F78:0B 16 16 16 92 92 16 16 <FA>
3F80:16 16 54 54 54 54 4C 4C <AF>
3F88:4C 56 56 56 56 16 16 16 <EE>
3F90:16 16 92 95 92 16 09 0A <33>
3F98:0B 16 95 92 95 95 0C 4C <5A>
3FA0:0E 16 16 92 92 92 92 16 <92>
3FA8:16 16 54 54 54 54 4C 4C <36>
3FB0:4C 56 56 56 56 16 16 16 <67>
3FB8:95 92 95 92 92 16 0C 4C <41>
3FC0:0E 16 95 92 95 92 0F 10 <AF>
3FC8:11 16 16 92 92 92 92 16 <48>
3FD0:16 16 54 54 54 54 4C 4C <5F>
3FD8:4C 56 56 56 56 16 16 16 <3F>
3FE0:92 95 92 95 16 16 0F 10 <17>
3FE8:11 16 95 92 92 95 16 16 <4D>
3FF0:16 16 16 16 16 16 16 16 <4A>
3FF8:16 16 54 54 54 57 58 58 <56>
4000:58 59 56 56 16 51 52 52 <CB>
4008:52 52 52 52 52 53 16 51 <CA>
4010:52 52 52 52 52 52 52 53 <48>
4018:16 51 52 52 52 52 52 52 <11>
4020:52 53 16 51 53 16 16 16 <83>
4028:16 16 51 53 16 54 55 05 <2A>
4030:55 05 55 05 55 56 16 54 <F3>
4038:55 05 55 05 55 05 55 56 <57>
4040:16 54 55 05 55 05 55 05 <4A>
4048:55 56 16 54 55 53 16 16 <F6>
4050:16 51 55 56 16 54 05 05 <41>
4058:51 52 53 05 05 56 16 54 <41>
4060:05 05 51 52 53 05 05 56 <F2>
4068:16 54 05 05 51 52 53 05 <37>
4070:05 56 16 57 05 05 53 16 <E9>
4078:51 05 05 59 16 54 55 05 <4A>
4080:54 55 56 05 55 56 16 54 <FE>
4088:55 05 57 58 59 05 55 56 <33>
4090:16 54 55 05 54 55 56 05 <D4>
4098:55 56 16 16 57 55 05 52 <E4>
40A0:05 55 59 16 16 54 05 05 <75>
40A8:57 58 59 05 05 56 16 54 <67>
40B0:05 05 05 05 58 58 58 59 <0E>
40B8:16 54 05 05 57 58 59 05 <1B>
40C0:05 56 16 16 16 54 05 05 <8D>
40C8:05 56 16 16 16 54 55 05 <36>
40D0:55 05 55 05 55 56 16 54 <93>
40D8:55 55 57 55 53 16 16 16 <25>
40E0:16 54 55 05 55 05 55 05 <A7>
40E8:55 56 16 16 16 54 55 05 <FD>
40F0:55 56 16 16 16 54 05 56 <B6>
40F8:16 16 16 54 05 56 16 54 <82>
4100:05 56 16 57 05 53 16 16 <76>
4108:16 54 05 56 16 16 16 54 <29>
4110:05 56 16 16 51 05 05 58 <C7>
4118:05 05 53 16 16 54 55 56 <08>
4120:16 16 16 54 55 56 16 54 <BF>
4128:55 56 16 16 57 55 53 16 <D3>
4130:16 54 55 56 16 16 16 54 <1B>
4138:55 56 16 51 55 05 59 16 <48>
4140:57 05 55 53 16 54 05 56 <E9>
4148:16 16 16 54 05 56 16 54 <50>
4150:05 56 16 16 16 57 05 53 <76>
4158:16 54 05 56 16 16 16 54 <78>
4160:05 56 16 54 05 59 16 16 <BE>
4168:16 57 05 56 16 57 58 59 <97>
4170:16 16 16 57 58 59 16 57 <26>
4178:58 59 16 16 16 16 57 59 <21>
4180:16 57 58 59 16 16 16 57 <FF>
4188:58 59 16 57 59 16 16 16 <D9>
4190:16 16 57 59 16 16 16 16 <5A>
4198:16 16 16 16 16 16 16 16 <18>
41A0:16 16 16 16 16 16 16 16 <20>
41A8:16 16 16 16 16 16 16 16 <28>
41B0:16 16 16 16 FF FF 00 06 <A0>
41B8:7F 06 0A 0A 0A 0A 0A 0A <F3>
41C0:0A 0A 0A 00 01 02 03 04 <25>
41C8:05 06 07 08 0A 0A 0A 0A <42>
41D0:0A 0A 0A 0A 0A 0A 0A 0A <52>
41D8:0A 0A 0A 0A 0A 0A 00 01 <3D>
41E0:02 03 04 05 06 07 08 00 <51>
41E8:01 02 03 04 05 06 07 08 <62>
41F0:00 01 02 03 04 05 06 07 <6A>
41F8:08 00 01 02 03 04 05 06 <F6>
4200:07 08 00 01 02 03 04 05 <43>
4208:06 07 08 0A 0A 0A 0A 0A <89>
4210:0A 0A 0A 0A 00 01 02 03 <09>
4218:04 05 06 07 08 00 01 02 <35>
4220:03 04 05 06 07 08 00 01 <81>
4228:02 03 04 05 06 07 08 00 <9B>
4230:01 02 03 04 05 06 07 08 <AC>
4238:00 00 FF FF 00 B8 FF BB <28>
4240:95 95 A5 A5 A9 A9 AA AA <3C>
4248:56 56 5A 5A 6A 6A AA AA <30>
4250:AA AA 6A 6A 5A 5A 56 56 <1A>
4258:AA AA A9 A9 A5 A5 95 95 <45>
4260:AA AA AA AA AA AA AA AA <07>
4268:FF FF FF FF FF FF FF FF <EC>
4270:FF FF FF AA AA FF FF FF <05>
4278:55 65 56 55 99 55 56 65 <84>
4280:EB EB EB EB EB EB EB EB <88>
4288:55 55 55 55 55 55 56 56 <CC>
4290:55 55 69 68 AA AA AA AA <5C>
4298:55 55 55 55 15 15 85 85 <87>
42A0:5A 5A 6A 6A 6A 6A 5A 5A <E3>
42A8:AA AA AA 00 00 AA AA AA <88>
42B0:A1 A1 A8 A8 A8 A8 A1 A1 <2C>
42B8:56 56 55 55 55 55 55 55 <D8>
42C0:AA AA AA AA 68 68 51 55 <4B>
42C8:85 85 15 15 55 55 55 55 <E1>
42D0:F1 45 CF FF CC 33 F1 4F <43>
42D8:73 C7 3C 1C FC 70 F0 C0 <C2>
42E0:F3 D1 33 0C 3C 07 0D 03 <54>
42E8:CF 4D F3 CC 34 7F C4 00 <65>
42F0:55 55 55 55 55 55 55 55 <E1>
42F8:00 FF 55 AA AA 55 FF 00 <06>
4300:3F 0F 0F C3 C3 F0 F0 FC <A9>
4308:95 A5 A5 69 69 5A 5A 56 <85>
4310:56 5A 5B 69 6D A5 B5 D5 <78>
4318:55 55 55 57 5F 5E 7B 78 <78>
4320:55 55 7D FF AA 33 BE 32 <0B>
4328:55 55 55 D5 F5 B5 2D ED <68>
4330:7B EC EF E3 EC EE E3 7B <D3>
4338:B3 FB 3C E3 FE 3C BB 3C <1E>
4340:ED 3B BB CB 3B BB 3B ED <00>
4348:78 7E 5E 5F 57 55 55 55 <F9>
4350:8E FF 3C AA FF 7D 55 55 <57>
4358:2D BD B5 F5 D5 55 55 55 <A6>
4360:FF FF FE FE FE F9 F9 F9 <84>
4368:FF 3F CE CE CE 79 79 79 <8F>
4370:FF 3F CF CF CF 73 73 73 <85>
4378:39 F9 D6 DA DA DE D7 DA <83>
4380:79 79 D6 BA BA D6 55 D7 <3E>
4388:70 7C DF B7 B7 DF 57 A7 <41>
4390:EE FF EE 3F C0 FF FF FF <75>
4398:EB AA C3 3C FF FF FF FF <4B>
43A0:BB FF BB FC 03 FF FF FF <8B>
43A8:FF FF FE FE FE F9 F9 F9 <CB>
43B0:FF FF 3E 3E 3E 49 49 69 <1F>
43B8:FF FF 3F 3F 3F 4F 4F 4F <5A>
43C0:E9 A9 95 95 9A 9E 97 9A <F5>
43C8:79 79 55 79 BA D6 55 D7 <D1>
43D0:63 58 58 78 B8 D8 58 A8 <7D>
43D8:A5 95 A5 2A C0 FF FF FF <04>
43E0:65 99 99 AA 00 FF FF FF <B5>
43E8:68 58 68 A3 0F FF FF FF <D0>
43F0:BB AE AB AA 6A 6A 6A 6A <AF>
43F8:BB EE BB EE BB AF AF AF <2C>
4400:BB EF BF FF FF FF FF FF <FA>
4408:5A 5A 5A 5A 56 56 56 56 <FF>
4410:AF AF AF AF AF AF AF AF <29>
4418:F5 F5 F5 F5 D5 D5 D5 D5 <C0>
4420:55 55 55 55 55 55 55 55 <A4>
4428:AF AF AF AF 6D 6D 6D 6D <13>
4430:55 55 55 55 55 55 55 55 <94>
4438:FF F5 ED EB EA EA EA EA <BE>
4440:FF 55 55 55 D7 BF AF AF <3F>
4448:FF 5F 7F FF FF FF FF FF <FC>
4450:EA EA EA EA EA EA EA EA <D8>
4458:AF AF AF AF AF AF AF AF <01>
4460:FF FF FF FF FF FF FF FF <E8>
4468:EA EA EA EA EB EC F0 FF <42>
4470:AF AF BF C3 00 00 00 FF <3F>
4478:FF FF FF FF FF 3F 0F FF <9D>
4480:55 55 55 5A 6A 69 69 A5 <84>
4488:55 55 AA AA 55 55 7D FF <9F>
4490:55 55 55 A5 A9 69 5B 5F <FB>
4498:A5 A5 A5 A5 A5 A5 A5 A5 <B5>
44A0:FF FE FA FA FA FA FA EA <9B>
44A8:5F 5F 5F 5F 5F 5F 5F 5F <0D>
44B0:A5 69 69 6D 5F 55 55 55 <34>
44B8:AA 69 55 55 FF FF 55 55 <61>
44C0:5F 7D 7D FD F5 55 55 55 <71>
44C8:FF EA EA E9 E9 E9 E9 E9 <BB>
44D0:FF AA AA 55 FF FF FF FF <DD>
44D8:FF AB A3 43 C3 C3 C3 C3 <8E>
44E0:E9 E9 E9 E9 E9 E9 E9 E9 <68>
44E8:FF FF EB BC BC C3 FF FF <B6>
44F0:C3 C3 C3 C3 C3 C3 C3 C3 <8A>
44F8:E9 E9 E9 E9 E9 E8 E0 FF <01>
4500:FF FF FF FF FF 00 00 FF <90>
4508:C3 C3 C3 C3 C3 03 03 FF <F8>
4510:AA AA AA AF BA B9 E6 9A <94>
4518:AA AA FF FF FF 7D BD BD <54>
4520:AA AA AA FA AE 6E 5B 56 <93>
4528:DA F6 FD FF FF FD F5 D5 <57>
4530:EB AA 96 96 96 96 AA EB <E7>
4538:57 5F 7F FF FF 7F DF D7 <70>
4540:95 E5 B9 BA AF AA AA AA <22>
4548:7F 7F 7D FF FF FF AA AA <62>
4550:F6 DB 6E AE FA AA AA AA <AD>
4558:FF FF FF FF FE F9 E6 96 <27>
4560:FF FF FF FF AA 55 AA 54 <49>
4568:FF FF FF FF BF 4F 53 54 <33>
4570:96 95 AA 02 FE FE FE FA <5E>
4578:00 55 55 59 59 59 AA 55 <47>
4580:54 54 40 4F 4F 4F 8F 6F <41>
4588:F9 EA 85 90 10 C5 F0 FF <25>
4590:65 AA 14 41 41 14 00 FF <5F>
4598:6F AB 52 06 04 53 0F FF <B5>
45A0:AA AE AB AE AB BA AE BA <80>
45A8:AA ED BB ED B7 ED B7 DD <44>
45B0:AA DE 7A DA 6E EE BE BE <D8>
45B8:AE BA AE BA AE B6 9E B6 <36>
45C0:AA 96 57 5F 5F 7F BE AA <6D>
45C8:BE BE BE B2 8E B2 8E B2 <99>
45D0:9E B6 9B BB AF AF BF AA <ED>
45D8:FF FC FF F0 FF CC F3 AA <86>
45E0:8E B2 2E E2 3A CA 32 AA <68>
45E8:AA AA AA AF B5 B6 B5 DA <DA>
45F0:AA AA FF 96 57 65 57 A6 <D6>
45F8:AA AA AA FA 5E 7E 57 77 <3D>
4600:D6 E6 D5 D9 F7 D5 DD D5 <33>
4608:A6 A5 59 97 55 DD 55 77 <DF>
4610:77 5F DF 5F 77 7F DF FF <68>
4618:F5 B7 BD B7 AF AA AA AA <FD>
4620:5D FF 5F FD FF FF AA AA <A9>
4628:77 FE 7E FE FA AA AA AA <15>
4630:00 00 00 00 00 00 00 00 <BC>
4638:00 00 00 00 00 00 00 00 <C4>
4640:FF FF E0 02 E1 02 00 A9 <E8>
4648:00 00 00 00 00 00 00 00 <F0>
```

»Arax« (Schluß)



Im Inneren des Computers

Um die Atari-Computer vernünftig programmieren zu können, braucht man die Entwicklungsunterlagen von Atari. Da diese nicht mehr erhältlich sind, haben wir für Sie die kompletten Hardware-Programmierunterlagen zusammengestellt.

Sicher kommt Ihnen folgende Szene bekannt vor: Sie sitzen vor Ihrem Computer und entwickeln Ihr neuestes Programm. An einer Stelle muß das Programm unbedingt die Funktionstasten abfragen, wobei Sie in Basic den Befehl »PEEK« verwenden. Nur, welche Adresse muß beim »PEEK«-Befehl angegeben werden? Also geht die Suche nach Unterlagen los, in denen die Adressen der Funktionstasten stehen. Ganz nach dem Motto: »Da war doch mal was in einer Zeitschrift...«, oder: »Ich hab da doch mal ein Buch gelesen...«.

Es gibt nun eine ganze Menge Literatur über die XL/XE-Computer. Leider ist in keinem Werk der gesamte Computer in allen Einzelheiten beschrieben. Das eine Buch behandelt zum Beispiel die Player-Missile-Grafik sehr ausführlich, dafür sind die Soundregister eher spärlich erklärt. Ein anderes behandelt diese genauer, verliert dafür aber kein Wort über die Display-List-Programmierung. Um vernünftig mit dem Computer arbeiten zu können, braucht man also immer mehrere Werke. Dabei blättert man sich die Finger wund, bis man eine brauchbare Beschreibung gefunden hat.

Atari veröffentlichte nun im Jahr 1979 für die alten 400er und 800er Computer sämtliche Programmierunterlagen. Darin enthalten waren das komplette dokumentierte Betriebssystem-Listing zusammen mit dessen Benutzerhandbuch sowie ein umfangreiches Werk, das sämtliche Hardwarebeschreibungen der Computer enthielt. Mit Hilfe dieser Unterlagen ließen sich professionelle Programme schreiben. Diese Unterlagen sind jedoch heute leider nicht mehr erhältlich. Einige Glückliche besitzen sie noch und hüten sie wie ihren Augapfel.

So kommt es, daß viele Programmierer gute Ideen haben, diese aber wegen mangelhafter Kenntnis über die Hardware der Computer nicht umsetzen können. Da Atari glücklicherweise mit den neuen Computern zu den alten Geräten so kompatibel wie möglich geblieben ist, lassen sich die Hardwareunterlagen der alten Geräte ohne größere Probleme auch für die XL/XE-Computer einsetzen.

Wir haben für Sie auf den folgenden Seiten diese Unterlagen zusammengestellt, um Ihnen ein umfassendes Nachschlagewerk in die Hand zu geben, und damit Sie fortan Ihren fantastischen Atari-Computer besser nutzen.

Die Übersicht ist in verschiedene Kategorien für die einzelnen Bereiche des Computers aufgeteilt. Fettgedruckt ist dann jeweils der Name des Registers und, falls vorhanden, der des entsprechenden Schattenregisters. Darunter befindet sich ein Kasten mit der Registeraufteilung. Ein graues Raster signalisiert ein nicht verwendetes Bit. Jedes benutzte Bit ist numeriert und wird im anschließenden Text erklärt. (hf)

Bildschirmsteuerung

PAL 53268 ($D014): Fernsehnorm

	B3	B2	B1	

D3	D2	D1	
0	0	0	PAL (europäisches Fernsehsystem)
1	1	1	NTSC (amerikanisches Fernsehsystem)

Mit diesem Byte stellt ein Programm fest, ob es auf einem amerikanischen oder deutschen XL/XE-Computer läuft.

VCOUNT 54283 ($D40B): Vertikaler Zeilenzähler

B7	B6	B5	B4	B3	B2	B1	B0

In dieser Adresse zählt der Computer die Bildschirmzeile mit, in der sich der Elektronenstrahl des Bildschirms gerade befindet. Zu beachten ist dabei, daß in VCOUNT nur jede zweite Bildschirmzeile gezählt wird.

WSYNC 54282 ($D40A): Warte auf horizontales Synchronsignal

Jeder Befehl des Mikroprozessors, in diese Speicherzelle zu schreiben, hält den Computer für die Dauer einer Bildschirmzeile an. Die Daten, die in dieses Register geschrieben werden, sind ohne Bedeutung.

DMACTL 54272 ($D400): DMA-Kontrollregister

	B5	B4	B3	B2	B1	B0

B5	schaltet den ANTIC ein
B4	wechselt die Auflösung der Player-Missile-Grafik. Wenn dieses Bit gelöscht ist, wird die zweizeilige Auflösung verwendet, sonst ist die einzeilige Auflösung eingeschaltet.
B3	befiehlt dem ANTIC, aus dem Speicher Byte-Folgen ab der Adresse, die mit »PMBASE« definiert wurde, auszulesen und diese Byte auf dem Bildschirm darzustellen (Player DMA)
B2	schaltet Missile DMA ein

B1	B0	Erklärung
0	0	Bildschirmgrafik ist abgeschaltet
0	1	Es wird ein schmaler Bildschirm mit 128 Punkten bei ANTIC-Mode 14 generiert
1	0	Es wird ein normal breiter Bildschirm generiert
1	1	Es wird ein überbreiter Bildschirm mit 192 Punkten beim ANTIC-Mode 14 generiert

GRACTL 53277 ($D01D): Grafik-Kontrollregister

	B2	B1	B0

B2	Bei gesetztem Bit wird ein Knopfdruck am Joystick zwischengespeichert
B1	schaltet die Player ein
B0	schaltet die Missiles ein

DLISTL 54274 ($D402): Low-Byte der Display-List-Adresse
Schattenregister: **SDLSTL** 560 ($230)

B7	B6	B5	B4	B3	B2	B1	B0

DLISTH 54275 ($D402): High-Byte der Display-List-Adresse
Schattenregister: **SDLSTH** 561 ($231)

B7	B6	B5	B4	B3	B2	B1	B0

Die Display-List ist eine Folge von Grafik-Instruktionen für den ANTIC-Coprozessor. Diese beiden Speicherstellen bestimmen, an welcher Stelle die Display-List steht. Man sollte die Speicherstellen nur mit ausgeschaltetem VBLANK-Interrupt verändern, da sonst ein Interrupt auftreten kann, während nur eine der beiden Speicherstellen verändert ist. In dem Fall sucht der ANTIC in einem völlig undefinierten Speicherbereich die Display-List, und der Bildschirm fängt an zu flackern.

CHBASE 54281 (D409): Startadresse des Zeichensatzes
Schattenregister: **CHBAS** 756 ($2F4)
40-Zeichen-Modus

B7	B6	B5	B4	B3	B2	

20-Zeichen-Modus

B7	B6	B5	B4	B3	B2	B1	

In CHBASE steht das höherwertige Byte der Adresse des Zeichensatzes. Je nach verwendeter Zeichensatzgrafik muß der Zeichensatz am Beginn einer 1-KByte- beziehungsweise 512-Byte-Grenze stehen. Die ersten 1 oder 2 Bit von CHBASE werden automatisch auf 0 gesetzt.

CHACTL 53273 ($D401): Steuerregister für den Zeichensatz
Schattenregister: CHART 755 ($2F3)

	B2	B1	B0

B2	Am Anfang einer Zeichensatzzeile wird dieses Bit abgefragt. Falls es gesetzt ist, werden alle Zeichen dieser Zeile auf den Kopf gestellt.
B1	ist nur im 40-Zeichen-Modus aktiv. Falls dieses Bit gesetzt ist, werden alle Zeichen, deren Code ein gesetztes siebtes Bit hat, invers dargestellt.
B0	ist nur im 40-Zeichen-Modus aktiv. Falls dieses Bit gesetzt ist, werden alle Zeichen, deren Code ein gesetztes siebtes Bit hat, gelöscht.

HSCROL 54276 ($D404): Horizontales Scrollregister

	B3	B2	B1	B0

Die Display-List-Zeile, in der die horizontale Scroll-Option eingeschaltet ist, wird zwischen 0 und 15 Bildschirmpunkte nach rechts verschoben. Zu beachten ist dabei, daß eine Verschiebung um einen Bildschirmpunkt zwei Punkten in Grafikstufe 8 entspricht.

Falls horizontales Scrolling eingeschaltet ist, werden auf dem Bildschirm mehrere Bytes für eine Zeile benötigt (siehe auch »DMACTL«). Eine Bildschirmzeile des schmalen Bildschirms benötigt dabei so viele Bytes wie eine Zeile des normalen Bildschirms. Eine Standardbildschirmzeile



benötigt so viele Bytes wie eine erweiterte Bildschirmzeile. Bei der erweiterten Bildschirmzeile erhöht sich bei eingeschaltetem horizontalen Scrolling die Anzahl der Bytes pro Zeile nicht.

VSCROL 54277 ($D405): Vertikales Scrollregister
16-Zeilen ANTIC-Befehle

	B3	B2	B1	B0

8-Zeilen ANTIC-Befehle

	B2	B1	B0

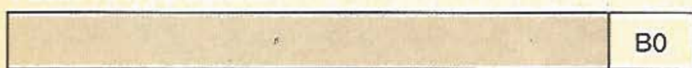
4-Zeilen ANTIC-Befehle

	B1	B0

2-Zeilen ANTIC-Befehle

	B0

1-Zeile ANTIC-Befehle



Sämtliche Bildschirmzeilen, in der die vertikale Scroll-Option eingeschaltet ist, werden um die Anzahl der Zeilen, die in VSCROL definiert ist, nach oben geschoben. Die Zeilen werden ab dem ersten Display-List-Befehl, in dem die Option nicht eingeschaltet ist, nicht mehr verschoben.

PRIOR 53275 ($D01B): Priorität der Player/Missiles und Grafiken
Schattenregister: GPRIOR 623 ($26F)

B7	B6	B5	B4	B3	B2	B1	B0

B7	B6	
0	0	normale Grafik
0	1	Wie in Grafikstufe 9 stehen 1 Farbe in 16 Helligkeitsstufen zur Verfügung. Die Farbe wird mit Register »COLBK« definiert.
1	0	9 frei definierbare Farben stehen wie in Grafikstufe 10 zur Verfügung. Die Farben werden mit den Registern »COLPM1« bis »COLPM3« sowie »COLPF0« bis »COLPF3« definiert. Die Hintergrundfarbe wird mit »COLPM0« festgelegt.
1	1	Wie in Grafikstufe 11 stehen 16 Farben einer Helligkeit zu Verfügung. Die Helligkeit wird mit »COLBK« festgelegt.

Alle drei Stufen funktionieren nur mit einem Computer, der einen GTIA eingebaut hat. Ältere 400er und 800er Modelle hatten noch einen CTIA, bei dem diese Grafikstufen nicht funktionieren. Bei diesen Grafikstufen werden jeweils 4 Bildschirmpunkte des ANTIC-Modes 2, 3 oder 15 genommen und zu einer Farbe zusammengefaßt. Deshalb haben diese Bits auch nur in diesen Grafikstufen eine Wirkung. ANTIC-Mode 2 entspricht Grafikstufe 0 und ANTIC-Mode 15 Grafikstufe 8. ANTIC-Mode 3 ist im Basic und im Betriebssystem nicht eingebaut.

B5 — Multi-Color Player-Bit. Falls dieses gesetzt ist, werden die Farb-Bits von Player 0 und Player 1 sowie von Player 2 und Player 3 mit Exklusiv-Oder verknüpft. Auf diese Weise lassen sich in den Players 3 Farben darstellen.

B4 — 5. Player-Bit. Mit diesem Bit werden die Missiles 0 bis 3 zu einem Player zusammengefaßt. Die Farbe dieses Players wird mit »COLPF3« bestimmt.

B3 - B0 — Prioritäten der Player. Je nach dem gesetzten Bit wird ein Player von einem anderen oder von der Bildschirmgrafik verdeckt. Die Bits haben folgende Prioritätsreihenfolgen:

B3=1	B2=1	B1=1	B0=1
F0	PF0	P0	P0
F1	F1	P1	P1
P0	F2	F0	P2
P1	F3+P5	F1	P3
P2	P0	F2	F0
P3	P1	F3+P5	F1
F2	P2	P2	F2
F3+P5	P3	P3	F3+P5
BAK	BAK	BAK	BAK

»P« bedeutet Player und »F« Farbe. Die obersten Tabelleneinträge sind dabei jeweils vor den untersten auf dem Bildschirm zu sehen. Falls Bit 3 gesetzt ist, verdeckt Player 3 also Farbe 2, während bei gesetztem Bit 2 Farbe 2 Player 3 verdeckt. Sind mehrere der Bits 0 bis 3 gesetzt, so wird auf dem Bildschirm der Bildschirmpunkt auf schwarz geschaltet, bei dem beide Fälle erfüllt sind.

COLPF0 53270 ($D016): Farbregister
Schattenregister: **COLOR0** 708 ($2C4)

COLPF1 53271 ($D017)
Schattenregister: **COLOR1** 709 ($2C5)

COLPF2 53272 ($D018)
Schattenregister: **COLOR2** 710 ($2C6)

COLPF3 53273 ($D019)
Schattenregister: **COLOR3** 711 ($2C7)

COLBK 53274 ($D01A)
Schattenregister: **COLOR4** 712 ($2C8)
Mit B7 und B6 des Registers »PRIOR« gelöscht:

B7	B6	B5	B4	B3	B2	B1	

Mit mindestens einem von B7 oder B6 des Registers »PRIOR« gesetzt:

B7	B6	B5	B4	B3	B2	B1	B0

B3 - B0 — bestimmen die Helligkeit einer Farbe in 8 beziehungsweise 16 Stufen. B7 bis B4 bestimmen den Farbton.

B7	B6	B5	B4	Farbton
0	0	0	0	Grau (von Schwarz bis Weiß)
0	0	0	1	Dunkelorange
0	0	1	0	Rotorange
0	0	1	1	Rot
0	1	0	0	Violett
0	1	0	1	Violettblau
0	1	1	0	Blauviolett
0	1	1	1	Veilchenblau
1	0	0	0	Blau
1	0	0	1	Königsblau
1	0	1	0	Grün
1	0	1	1	Sattgrün
1	1	0	0	Grüngelb
1	1	0	1	Olive
1	1	1	0	Braunorange
1	1	1	1	Dunkelorange (wie Kombination 0001)

Player und Missiles

PMBASE 54279 ($D407): Startadresse des Player-Missile-Speicherbereichs
Bei einzeiliger Auflösung

B7	B6	B5	B4	B3	

Bei zweizeiliger Auflösung

B7	B6	B5	B4	B3	B2	

Der Speicherbereich der Player-Missiles muß je nach Auflösung an einer 2-KByte- beziehungsweise 1-KByte-Grenze liegen.

VDELAY 54276 ($D01C): Verschiebt die Player-Missiles um eine Bildschirmzeile nach unten

B7	B6	B5	B4	B3	B2	B1	B0

B7 verschiebt Player 3
B6 verschiebt Player 2
B5 verschiebt Player 1
B4 verschiebt Player 0
B3 verschiebt Missile 3
B2 verschiebt Missile 2
B1 verschiebt Missile 1
B0 verschiebt Missile 0

Diese Speicherstelle wird dazu benutzt, die Players oder Missiles bei zweizeiliger Auflösung eine Bildschirmzeile nach unten zu verschieben.

GRAFP0 53261 ($D00D): Grafik der Player
GRAFP1 53262 ($D00E)
GRAFP2 53263 ($D00F)
GRAFP3 53264 ($D010)

B7	B6	B5	B4	B3	B2	B1	B0

Das Bitmuster, das für die jeweiligen Players in die Speicherstellen übertragen wird, erscheint bei eingeschalteten Players, aber nicht aktiviertem Player-Missile-DMA (siehe »DMACTL«) auf dem Bildschirm.

GRAFM 53265 ($D011): Grafik der Missiles

B7	B6	B5	B4	B3	B2	B1	B0

B7 - B6 Bitmuster für Missile 3
B5 - B4 Bitmuster für Missile 2
B3 - B2 Bitmuster für Missile 1
B1 - B0 Bitmuster für Missile 0

Wie bei »GRAFP0« bis »GRAFP3« erscheint das in diese Speicherstellen geschriebene Bitmuster für die Missiles auf dem Bildschirm, wenn die Missiles eingeschaltet, der Player-Missile-DMA aber ausgeschaltet ist (siehe »DMACTL«).

SIZEP0 53256 ($D008): Breite der Player
SIZEP1 53257 ($D009)
SIZEP2 53258 ($D00A)
SIZEP3 53259 ($D00B)

	B1	B0

B1	B0	Breite
0	0	normale Breite, jedes Bit des Players hat auf dem Bildschirm die Breite zweier Grafikstufe-8-Punkte
0	1	doppelte Breite der Bildschirmpunkte
1	0	normale Breite
1	1	vierfache Breite

Mit diesem Byte lassen sich die Bildschirmpunkte eines Players um das Zwei- oder Vierfache dehnen.

SIZEM 53260 ($D00C)

B7	B6	B5	B4	B3	B2	B1	B0

B7 - B6 Breite von Missile 3
B5 - B4 Breite von Missile 2
B3 - B2 Breite von Missile 1
B1 - B0 Breite von Missile 0

Wie »SIZEP0« bis »SIZEP3« für jeden der vier Missiles.

HPOSP0 53248 ($D000): Horizontale Position des Players
HPOSP1 53249 ($D001)
HPOSP2 53250 ($D002)
HPOSP3 53251 ($D003)

B7	B6	B5	B4	B3	B2	B1	B0

Mit dieser Adresse wird die horizontale Position des jeweiligen Players festgelegt. Der Wert 48 ($30) entspricht dabei dem linken Rand eines normal breiten Bildchirms, 208 ($D0) dem rechten Rand.

HPOSM0 53252 ($D004): Horizontale Position des Missiles
HPOSM1 53253 ($D005)
HPOSM2 53254 ($D006)
HPOSM3 53255 ($D007)

B7	B6	B5	B4	B3	B2	B1	B0

Bestimmt die horizontale Position der Missiles auf dem Bildschirm. Die Werte entsprechen denen von »HPOSP0« bis »HPOSP3«.

COLPM0 53266 ($D012): Farbe der Player-Missiles
Schattenregister: **PCOLR0** 704 ($2C0)

COLPM1 53267 ($D013)
Schattenregister: **PCOLR1** 705 ($2C1)

COLPM2 53268 ($D014)
Schattenregister: **PCOLR2** 706 ($2C2)

COLPM3 53269 ($D015)
Schattenregister: **PCOLR3** 707 ($2C3)

B7	B6	B5	B4	B3	B2	B1	

Diese Register bestimmen die Farben der Player (siehe auch »COLPF0« bis »COLPF3« und »COLBK«).

M0PF 53248 ($D000): Kollision zwischen Missile und Bildschirmgrafik
M1PF 53249 ($D001)
M2PF 53250 ($D002)
M3PF 53251 ($D003)

	B3	B2	B1	B0

B3 entspricht COLPF3
B2 entspricht COLPF2
B1 entspricht COLPF1
B0 entspricht COLPF0

Diese vier Register signalisieren eine Berührung zwischen einer Bildschirmgrafik und einer Missile. Welche Farbe die Missile berührt hat, bestimmt jeweils eins der vier Bits. Die Bits bleiben auch nach der Berührung gesetzt.

P0PF 53252 ($D004): Kollision zwischen Player und Bildschirmgrafik
P1PF 53253 ($D005)
P2PF 53254 ($D006)
P3PF 53255 ($D007)

	B3	B2	B1	B0

M0PL 53256 ($D008): Kollision zwischen Missile und Player
M1PL 53257 ($D009)
M2PL 53258 ($D00A)
M3PL 53259 ($D00B)

	B3	B2	B1	B0

P0PL 53260 ($D00C): Kollision zweier Player
P1PL 53261 ($D00D)
P2PL 53262 ($D00E)
P3PL 53263 ($D00F)

	B3	B2	B1	B0

HITCLR 53278 ($D01E): Löscht alle Bit der Kollisionsregister.

Da alle Kollisionsregister nach einer entsprechenden Berührung mit dem Objekt ihre Werte behalten, lassen sie sich mit diesem Register wieder löschen. Die Daten, die in dieses Register geschrieben werden, sind ohne Belang.

Tonerzeugung

AUDCTL 53768 ($D208): Sound-Kontrollregister

B7	B6	B5	B4	B3	B2	B1	B0

B7 Der 17-Bit-Zufallszahlenzähler wird in einen 9-Bit-Zähler verwandelt
B6 Tonkanal 1 wird mit 1,79 MHz getaktet
B5 Tonkanal 3 wird mit 1,79 MHz getaktet
B4 Tonkanal 2 wird mit der Frequenz von Tonkanal 1 getaktet
B3 Tonkanal 4 wird mit der Frequenz von Tonkanal 3 getaktet
B2 Ein Hochpaßfilter mit der Eckfrequenz von Tonkanal 3 wird in Tonkanal 1 eingefügt
B1 Ein Hochpaßfilter mit der Eckfrequenz von Tonkanal 4 wird in Tonkanal 2 eingefügt
B0 Alle Tonkanäle werden nicht mit 64 kHz, sondern mit 15 kHz getaktet

Mit diesem Register lassen sich verschiedene Klangfarben eines Tons erzeugen. Am besten probiert man mit den einzelnen Bits herum, um zu hörenswerten Ergebnissen zu kommen.

Die oben gegebenen Frequenzen sind Annäherungswerte. Die exakten Frequenzwerte lauten folgendermaßen:

angenäherte Frequenz	exakte Frequenz
1,79 MHz	1,78979 MHz
64 KHz	63,921 KHz
15 KHz	15,9803 KHz

Eine ausgegebene Frequenz wird nach folgender Formel berechnet:

Ausgangsfrequenz = Eingangsfrequenz / zweifacher Registerwert + 1

Der Registerwert entspricht dem Wert, der in die einzelnen Register »AUDF1« bis »AUDF4« geschrieben wird. Falls eine höhere Taktfrequenz benutzt wird, muß eine andere Formel zur Berechnung der Tonfrequenz verwendet werden:

Ausgangsfrequenz = Eingangsfrequenz / 2* (Registerwert + O)

Wenn B3 oder B4 von »AUDCTL« 1 sind, wird »O« in der Formel zu 7, sonst wird »O« zu 4.

AUDF1 53760 ($D200): Tonfrequenz
AUDF2 53762 ($D202)
AUDF3 53764 ($D204)
AUDF4 53766 ($D206)

B7	B6	B5	B4	B3	B2	B1	B0

In dieses Register wird ein Wert zwischen 0 und 255 ($00 - $FF) geschrieben, der die Frequenz eines Tons eines der vier Tonkanäle bestimmt.

AUDC1 53761 ($D201): Lautstärke und Verzerrung des Tons
AUDC2 53763 ($D203)
AUDC3 53765 ($D205)
AUDC4 53767 ($D207)

B7	B6	B5	B4	B3	B2	B1	B0

B7	B6	B5	B4	Erklärung
0	0	0	0	Rauschen durch 5- und 17-Bit-Zufallsgenerator
0	0	1	0	Rauschen mit 5-Bit-Zufallsgenerator
0	1	0	0	Rauschen mit 4- und 5-Bit-Zufallsgenerator
0	1	1	0	Rauschen mit 5-Bit-Zufallsgenerator (wie Kombination 0010)
1	0	0	0	Rauschen mit 17-Bit-Zufallsgenerator
1	x	1	0	Reiner Ton
1	1	0	0	Rauschen mit 4-Bit-Zufallsgenerator
x	x	x	1	Kein Ton, nur Lautstärkewerte

B3 bis B0 enthalten einen Wert zwischen 0 und 15, der die Lautstärke des Tons bestimmt. Mit diesen Registern werden, ähnlich dem »SOUND«-Befehl in Basic, die Verzerrung sowie die Lautstärke eines Tons festgelegt.

Notenwerte		
Notenname	Dezimalwert	Hexadezimalwert
C4	29	$1D
H3	31	$1F
A#3	33	$21
A3	35	$23
G#3	37	$25
G3	40	$28
F#3	42	$2A
F3	45	$2D
E3	47	$2F
D#3	50	$32
D3	53	$35



Notenname	Dezimalwert	Hexadezimalwert
C#3	57	$39
C3	60	$3C
H2	64	$40
A#2	68	$44
A2	72	$48
G#2	76	$4C
G2	81	$51
F#2	85	$55
F2	91	$5B
E2	96	$60
D#2	102	$66
D2	108	$6C
C#2	114	$72
C2	121	$79
B1	128	$80
A#1	136	$88
A1	144	$90
G#1	153	$99
G1	162	$A2
F#1	173	$AD
F1	182	$B6
E1	193	$C1
D#1	204	$CC
D1	217	$D9
C#1	230	$E6
C1	243	$F3

RANDOM 53770 ($D20A): Zufallszahlengenerator

B7	B6	B5	B4	B3	B2	B1	B0

Aus dieser Speicherstelle lassen sich Zufallszahlen in einem Bereich zwischen 0 und 255 auslesen. B7 von »AUDCTL« verändert den Strom der Zufallszahlen.

Interruptsteuerung

NMIEN 54286 ($D40E): Schaltet NMIs ein und aus

B7	B6	

B7 Display-List-Interrupts ermöglichen. Wenn dieses Bit gesetzt wird, kann vom ANTIC aus ein Display-List-Interrupt ausgelöst werden.
B6 VBLANK-Interrupt ermöglichen. Falls dieses Bit gesetzt wird, löst der ANTIC bei jedem neuen Fernsehbild einen Interrupt aus.

NMIST 54287 ($D40F): Status der NMIs

B7	B6	B5	

B7 Falls dieses Bit gesetzt ist, wurde vom ANTIC ein Display-List-Interrupt ausgelöst
B6 Falls dieses Bit gesetzt ist, wurde vom ANTIC ein VBLANK-Interrupt ausgelöst
B5 Dieses Bit signalisiert, daß auf der Tastatur der alten 400er- und 800er-Computer die <SYSTEM RESET>-Taste gedrückt wurde. Da bei den XL/XE-Computern mit der <RESET>-Taste ein echter Reset des Mikroprozessors ausgelöst wird, hat dieses Bit seine Bedeutung verloren.

NMIRES 54287 ($D40F): Lösche Bit in »NMIST«

Sobald der Mikroprozessor einen Schreibbefehl in diese Adresse ausführt, werden sämtliche Bits von »NMIST« gelöscht. Die geschriebenen Daten sind dabei ohne Belang.

IRQEN 53774 ($D20E): Ermöglicht IRQs
Schattenregister: **POKMSK** 16 ($10)

B7	B6	B5	B4	B3	B2	B1	B0

B7 Es wird ein Interrupt ausgelöst, sobald die <BREAK>-Taste gedrückt wird
B6 löst einen Interrupt aus, sobald irgendeine Taste der Tastatur gedrückt wird (siehe auch »CONSOL«, »KBCODE« und »SKSTAT«)
B5 ermöglicht einen Interrupt, sobald ein Byte am seriellen Port eingelesen wurde
B4 ermöglicht, von einem angeschlossenen Peripheriegerät Daten anzufordern
B3 löst einen Interrupt aus, sobald alle Bit eines Byte zu einem angeschlossenem Peripheriegerät geschickt wurden
B2 Zeitgeber-4-Interrupt einschalten
B1 Zeitgeber-2-Interrupt einschalten
B0 Zeitgeber-1-Interrupt einschalten

IRQST 53774 ($D20E): Zeigt an, welcher Interrupt ausgelöst wurde

B7	B6	B5	B4	B3	B2	B1	B0

B7 Die <BREAK>-Taste hat einen Interrupt ausgelöst
B6 Eine Taste der Tastatur hat einen Interrupt ausgelöst
B5 Ein Byte wurde fehlerfrei vom seriellen Prot gelesen
B4 Ein Peripheriegerät hat ein Byte angefordert
B3 Ein Byte wurde komplett an ein Peripheriegerät übertragen
B2 Zeitgeber 4 hat einen Interrupt ausgelöst
B1 Zeitgeber 2 hat einen Interrupt ausgelöst
B0 Zeitgeber 1 hat einen Interrupt ausgelöst

Alle diese Interrupts werden durch einen IRQ am Mikroprozessor ausgelöst. Über dieses Register stellt ein Programm fest, welcher der acht IRQs ausgelöst wurde.

***STIMER** 53769 ($D209): Starte Timer*

Über dieses Register werden die Timer des Computers gestartet. Als Timer fungieren die Register »AUDF1«, »AUDF2« und »AUDF4«. Erreichen die Timer den Wert 0, und sind die entsprechenden Bits im »IRQEN«-Register gesetzt, wird ein IRQ ausgelöst. Der Wert, der in dieses Register geschrieben wird, ist ohne Bedeutung.

Tastatur

CONSOL 53279 ($D01F): Funktionstasten

	B3	B2	B1	B0

B3 Tastaturklick. In den alten 400er- und 800er-Computern war ein Lautsprecher eingebaut. Dieser ließ sich mit diesem Bit ansprechen. Schreibt man hier also in

schneller Reihenfolge eine 0 und eine 1 hinein, vernimmt man einen Ton aus dem Lautsprecher. Bei den XL/XE-Computern wird kein Lautsprecher eingebaut. Dafür hört man den Tastaturklick aus dem Lautsprecher des Fernsehers oder Monitors.
B2 Die <OPTION>-Taste wird gedrückt
B1 Die <SELECT>-Taste wird gedrückt
B0 Die <START>-Taste wird gedrückt.

Mit diesem Register werden die Funktionstasten abgefragt. Normalerweise sind die Bits B0 bis B2 auf 1 gesetzt. Erst wenn eine Taste gedrückt wird, ist das entsprechende Bit gelöscht.

KBCODE 53769 ($D209): Tastaturcode
Schattenregister: **CH** 764 ($2FC)

B7	B6	B5	B4	B3	B2	B1	B0

D7 <CONTROL>-Taste
D6 <SHIFT>-Taste

Die restlichen 6 Bit werden nach folgendem Schema aufgeschlüsselt:

Tastaturcode	Tastaturkappe	ASCII	<SHIFT>	<CONTROL>
$00	<L>	$6C	$4C	$0C
$01	<J>	$6A	$4A	$0A
$02	<;>	$3B	$3A	$7B
$03	<F1>	-	-	-
$04	<F2>	-	-	-
$05	<K>	$6B	$4B	$0B
$06	<+>	$2B	$5C	$1E
$07	<*>	$2A	$5E	$1F
$08	<O>	$6F	$4F	$0F
$09	-	-	-	-
$0A	<P>	$70	$50	$10
$0B	<U>	$75	$55	$15
$0C	<RETURN>	$9B	$9B	$9B
$0D	<I>	$69	$49	$09
$0E	<->	$2D	$5F	$1C
$0F	<=>	$3D	$7C	$1D
$10	<V>	$76	$56	$16
$11	<HELP>	-	-	-
$12	<C>	$63	$43	$03
$13	<F3>	-	-	-
$14	<F4>	-	-	-
$15	<B>	$62	$42	$02
$16	<X>	$78	$58	$18
$17	<Z>	$7A	$5A	$1A
$18	<4>	$34	$24	-
$19	-	-	-	-
$1A	<3>	$33	$23	*
$1B	<6>	$36	$26	-
$1C	<ESC>	$1B	$1B	$1B
$1D	<5>	$35	$25	-
$1E	<2>	$32	$22	Beep
$1F	<1>	$31	$21	*
$20	<,>	$2C	$5B	$00
$21	<SPACE>	$20	$20	$20
$22	<.>	$2E	$5D	$60
$23	<N>	$6E	$4E	$0E
$24	-	-	-	-
$25	<M>	$6D	$4D	$0D
$26	</>	$2F	$3F	-
$27	<INVERS>	*	*	*
$28	<R>	$72	$52	$12
$29	-	-	-	-
$2A	<E>	$65	$45	$05
$2B	<Y>	$79	$59	$19
$2C	<TAB>	$7F	$9F	$9E
$2D	<T>	$74	$54	$14
$2E	<W>	$77	$57	$17
$2F	<Q>	$71	$51	$11
$30	<9>	$39	$28	-
$31	-	-	-	-
$32	<0>	$30	$29	-
$33	<7>	$37	$27	-
$34	<BACK SPACE>	$7E	$9C	$FE
$35	<8>	$38	$40	-
$36	< < >	$3C	$7D	$7D
$37	< > >	$3E	$9D	$FF
$38	<F>	$66	$46	$06
$39	<H>	$68	$48	$08
$3A	<D>	$64	$44	$04
$3B	-	-	-	-
$3C	<CAPS>	*	*	*
$3D	<G>	$67	$47	$07
$3E	<S>	$73	$53	$13
$3F	<A>	$61	$41	$01

Die mit einem »*« gekennzeichneten Tasten erhalten keinen Tastaturcode, sondern werden vom Betriebssystem für spezielle Zwecke benötigt. Bit B7 und B6 lassen sich nur abfragen, wenn die <SHIFT> oder <CONTROL>-Taste zusammen mit einer anderen Taste gedrückt wird.

Joystickanschlüsse

PORTA 54016 ($D300)
Schattenregister: **STICK0** 632 ($278): Joystickanschluß 1 und 2
STICK1 633 ($279)
PTRIG0 636 ($27C)
PTRIG1 637 ($27D)
PTRIG2 638 ($27E)
PTRIG3 639 ($27F)
Für Joysticks

B7	B6	B5	B4	B3	B2	B1	B0

B7 Joystick 2 nach rechts gedrückt
B6 Joystick 2 nach links gedrückt
B5 Joystick 2 nach hinten gedrückt
B4 Joystick 2 nach vorne gedrückt
B3 Joystick 1 nach rechts gedrückt
B2 Joystick 1 nach links gedrückt
B1 Joystick 1 nach hinten gedrückt
B0 Joystick 1 nach vorne gedrückt

Für Drehregler

B7	B6			B3	B2		

B7 Feuerknopf Drehregler 3
B6 Feuerknopf Drehregler 2
B3 Feuerknopf Drehregler 1
B2 Feuerknopf Drehregler 0

Mit diesem Register lassen sich die Joysticks 1 und 2 sowie die Drehregler 1, 2, 3 und 4 abfragen, sofern in »PACTL« B2 gesetzt ist.

Falls B2 in »PACTL« gelöscht ist, wird mit dieser Adresse das Richtungsregister angesprochen.

B7	B6	B5	B4	B3	B2	B1	B0

Jedes Bit in diesem Register schaltet am Joystickport den betreffenden Pin als Eingang oder Ausgang. Ein gelöschtes Bit schaltet den Pin als Eingang und ein gesetztes Bit schaltet einen Ausgang.

PACTL 54018 ($D302): Port A Kontrollregister

B7	B6	B5	B4	B3	B2	B1	B0

B7 Dieses Bit läßt sich nur lesen. Es zeigt den Zustand des »Proceed«-Pins des seriellen Bus an. Wenn das Bit gesetzt ist, wurde von einem Peripheriegerät ein Interrupt ausgelöst. Das Bit läßt sich durch Lesen von »PORTA« wieder löschen.
B6 ist immer 0
B5 ist immer 1
B4 ist immer 1
B3 kontrolliert den Motor vom Kassettenrecorder. Ist das Bit gelöscht, so ist der Motor eingeschaltet.
B2 schaltet bei »PORTA« zwischen dem Daten- und dem Richtungsregister hin und her. Wird das Bit gesetzt, so ist »PORTA« als Datenregister geschaltet.
B1 ist immer 0
B0 bestimmt, ob über den »Proceed«-Pin am seriellen Port ein Interrupt auftreten darf. Wenn das Bit gesetzt ist, wird ein Interrupt akzeptiert.

PORTB 54017 ($D301): Banklogik

B7	B6	B5	B4	B3	B2	B1	B0

B7 Wenn dieses Bit gesetzt ist, wird im Speicher zwischen $5000 und $57FF anstelle des RAM das Selbsttest-ROM eingeblendet
B6 ist nicht benutzt.
B5 hat nur beim 130XE eine Bedeutung. Wenn dieses Bit gesetzt ist, hat der ANTIC auf die zusätzlichen 64 KByte RAM Zugriff.
B4 hat nur beim 130XE eine Bedeutung. Wenn dieses Bit gesetzt ist, hat die CPU auf die zusätzlichen 64 KByte RAM Zugriff.
B2–B3 haben ebenfalls nur beim 130XE eine Bedeutung. Mit ihnen wählt man eine der zusätzlichen vier RAM-Bänke im Bereich von $4000 bis $7FFF aus.

B3	B2	
0	0	RAM-Bank 0
0	1	RAM-Bank 1
1	0	RAM-Bank 2
1	1	RAM-Bank 3

B1 schaltet im Speicherbereich zwischen $A000 und $BFFF das Atari-Basic-ROM ein. Wird dieses Bit gesetzt, so wird das ROM eingeblendet.
B0 schaltet im Bereich zwischen $C000 und $CFFF sowie zwischen $D800 bis $FFFF das Betriebssystem-ROM ab und blendet das darunterliegende RAM ein. Wenn B0 gelöscht ist, wird das ROM ausgeblendet.

Bei den alten Atari 400er und 800er Computern werden hier die Joysticks und Drehregler von Port 3 und 4 abgefragt (siehe PORTA).
Schattenregister: **STICK2** 634 ($27A)
STICK3 635 ($27B)
PTRIG4 640 ($280)
PTRIG5 641 ($281)
PTRIG6 642 ($282)
PTRIG7 643 ($283)

PBCTL 54019 ($D303): Port B Kontrollregister

B7	B6	B5	B4	B3	B2	B1	B0

B7 Dieses Bit läßt sich nur lesen. Wenn es gesetzt ist, wurde von einem Peripheriegerät am »Interrupt«-Pin des seriellen Bus ein Interrupt ausgelöst. Das Bit läßt sich durch Lesen von »PORTB« wieder löschen.
B6 ist immer 0
B5 ist immer 1
B4 ist immer 1
B3 zeigt den Zustand des »Command»-Pin am seriellen Bus an
B2 wechselt bei »PORTB« zwischen dem Daten- und dem Richtungsregister. Wenn das Bit gesetzt ist, ist am »PORTB«-Register das Datenregister ansprechbar.
B1 ist immer 0
B0 ermöglicht einen Interrupt am »Interrupt»-Pin des seriellen Bus

POT0 53760 ($D200): Drehreglereingänge
Schattenregister: **PADDLE0** 624 ($270)

POT1 53761 ($D201)
Schattenregister: **PADDLE1** 625 ($271)

POT2 53762 ($D202)
Schattenregister: **PADDLE2** 626 ($272)

POT3 53763 ($D203)
Schattenregister: **PADDLE3** 627 ($273)

POT4 53764 ($D204)
Schattenregister: **PADDLE4** 628 ($274)

POT5 53765 ($D205)
Schattenregister: **PADDLE5** 629 ($275)

POT6 53766 ($D206)
Schattenregister: **PADDLE6** 630 ($276)

POT7 53767 ($D207)
Schattenregister: **PADDLE7** 631 ($277)

B7	B6	B5	B4	B3	B2	B1	B0

Mit diesen Registern wird der Wert der Drehregler abgelesen. Der Wert liegt immer zwischen 0 und 228 ($0 – $E4). Die letzten vier Register haben bei den XL/XE-Computern keine Bedeutung, da bei diesen die Anschlüsse für die Drehregler nicht vorhanden sind. Der Wert in den einzelnen Registern ist erst nach 228 Fernsehzeilen auf dem Bildschirm gültig (siehe auch »ALLPOT« und »POTGO«).

ALLPOT 53768 ($D208): Gültigkeit der Drehreglerwerte

B7	B6	B5	B4	B3	B2	B1	B0

Dieses Register wird ausgelesen, um festzustellen, ob der Wert eines Drehreglers einen stabilen Zustand erreicht hat. B7 entspricht Drehregler 7 und B0 Drehregler 0. Ein gesetztes Bit zeigt einen gültigen Drehreglerwert an.

POTGO 53771 ($D20B): Drehregler abfragen

Mit diesem Register wird eine Abfragesequenz der Drehregler gestartet. Nach spätestens 228 Bildschirmzeilen ist die Sequenz beendet (siehe »ALLPOT«) und die Werte können aus »POT0« bis »POT7« gelesen werden.

TRIG0 53264 ($D010): Feuerknöpfe der Joysticks
Schattenregister: **STRIG0** 644 ($284)

TRIG1 53265 ($D011)
Schattenregister: **STRIG1** 645 ($285)

TRIG2 53266 ($D012)
Schattenregister: **STRIG2** 646 ($286)

TRIG3 53267 ($D013)
Schattenregister: **STRIG3** 647 ($287)

	B0

Diese Speicherstellen fragen die Feuerknöpfe der Joysticks ab. Immer wenn ein Knopf am Joystick gedrückt wurde, ist das Bit B0 gelöscht. Wenn in dem Register »GRACTL« B2 gesetzt ist, speichern die Register den Druck des Joystick-Knopfs. Zurückgesetzt werden sie, indem B2 von »GRACTL« gelöscht wird (siehe auch »GRACTL«).

PENH 54284 ($D40C): Lichtgriffel
Schattenregister: **LPENH** 564 ($234)

B7	B6	B5	B4	B3	B2	B1	B0

Dieses Register liest die horizontale Position des Lichtgriffels auf dem Bildschirm aus. Der Wert liegt zwischen 0 und 227 ($0 - $E3). Aufgrund eines Hardwarefehlers verändern sich die Werte in dem Register, wenn der Feuerknopf am Joystick gedrückt ist. Zuverlässig funktioniert der Lichtgriffel nur bei dem amerikanischen NTSC-Fernsehsystem.

PENV 54285 ($D40D): Lichtgriffel
Schattenregister: **LPENV** 565 ($235)

B7	B6	B5	B4	B3	B2	B1	B0

Dieses Register gibt die vertikale Position eines Lichtgriffels auf dem Bildschirm an (siehe auch »PENH«).

Serieller Port

SKCTL 53775 ($D20F): Steuerung serieller Port

B7	B6	B5	B4	B3	B2	B1	B0

B7 schaltet den Pin »seriell Output« vom Port auf Masse

B6	B5	B4	Erklärung
0	0	0	Ein- und Ausgabe von Daten werden mit einem externen Taktsignal an »Clock in« vom seriellen Port gesteuert
0	0	1	Ausgabe von Daten wird mit einem externen Takt an »Clock in« gesteuert. Bei der Dateneingabe übernimmt der Computer die Daten im Takt des Tonkanals 4.
0	1	0	Ein- und Ausgabe von Daten erfolgt im Takt des Tonkanals 4. Der Takt steht am »Clock out«-Pin des seriellen Port zur Verfügung.
0	1	1	-
1	0	0	Daten werden im Takt des Tonkanals 4 ausgegeben. Dateneingabe erfolgt über einen externen Taktgeber am »Clock in«-Pin des seriellen Port.
1	0	1	-
1	1	0	Datenausgabe erfolgt im Takt des Tonkanals 2. Dateneingabe erfolgt im Takt des Tonkanals 4, wobei der Takt am »Clock out«-Pin des seriellen Port zur Verfügung steht.
1	1	1	Datenausgabe erfolgt im Takt des Tonkanals 2, Dateneingabe im Takt des Tonkanals 4. Es steht kein Taktsignal zur Verfügung.

B3 setzt die Zweiton-Datenübertragung (wird beim Kassettenrecorder benutzt) anstelle einer digitalen Datenübertragung. Die zwei Töne werden mit Tonkanal 1 und 2 erzeugt.

B2 Mit diesem Bit wird die schnelle Abfrage der Drehregleranschlüsse eingeleitet. Anstelle von 228 Zeilen lassen sich die »POT«-Register schon nach zwei Bildschirmzeilen auslesen.

B1 schaltet die Tastatur-Abfrage ein

B1 schaltet die Tastatur-Entprellung ein

Wenn B0 und B1 gelöscht sind, führt der POKEY eine Selbsttest-Routine aus.

SKSTAT 53775 ($D20F): Status des seriellen Port

B7	B6	B5	B4	B3	B2	B1	B0

B7 Wenn dieses Bit gelöscht ist, wurde ein Byte von einem Peripheriegerät zum Computer gesendet, während dieser das letzte noch nicht verarbeitet hat

B6 Wenn dieses Bit gelöscht ist, wollte der Computer ein weiteres Byte senden, während das vorige noch nicht komplett zum Peripheriegerät geschickt wurde

B5 Dieses Bit zeigt bei gelöschtem Zustand an, daß auf der Tastatur zu schnell hintereinander Tasten gedrückt wurden

B4 ist direkt mit dem »Seriell In«-Pin des seriellen Port verbunden

B3 Wenn eine der beiden <SHIFT>-Tasten gedrückt sind, wird dieses Bit gelöscht

B2 Eine Taste auf der Tastatur ist immer noch gedrückt.

B1 zeigt bei gelöschtem Zustand an, daß der Computer gerade ein Byte von einem Peripheriegerät empfängt

B0 ist immer 1

SKRES 53770 ($D20A)

Wenn die CPU einen Schreibbefehl in dieses Register ausführt, werden alle Bits von »SKSTAT« gelöscht. Die Daten, die in das Register geschrieben werden, sind ohne Bedeutung.

SERIN 53774 ($D20E): Serielles Empfangsregister

B7	B6	B5	B4	B3	B2	B1	B0

Der Computer empfängt von einem Peripheriegerät seriell einzelne Bits eines Byte. Nach der Übertragung steht das übertragene Byte in diesem Register zur Verfügung.

SEROUT 53773 ($D20D): Serielles Senderegister

B7	B6	B5	B4	B3	B2	B1	B0

In dieses Register schreibt der Computer ein Byte, das seriell zu den Peripheriegeräten übertragen wird.

Mit diesen neun Seiten wäre der komplette Computer beschrieben. Wie auch anderswo macht nur die Übung den Meister. Sie sollten also ausgiebig mit den Registern experimentieren. Erst dann wir auch ersichtlich, welches Potential in dem kleinen Computer steckt. Wahrscheinlich werden Sie nach kurzer Zeit die Fähigkeiten Ihres Gerätes nicht mehr missen wollen. Gerade wenn Sie auf einen anderen Computer umsteigen, werden Ihnen sicherlich die vielen Funktionen der XL/XE-Computer fehlen. Auf der folgenden Seite sind noch einmal sämtliche Register sowie deren Schattenregister in alphabetischer Reihenfolge zusammengestellt.



Die Register in alphabetischer Reihenfolge

Hardware-Register		Adresse		Schattenregister	Adresse	
Name	Beschreibung	Hex	Dez	Name	Hex	Dez
ALLPOT	Abfrage ob alle Drehegler fertig gelesen	D208	53768			
AUDC1	Tonkanal 1 Kontrollregister	D201	53761			
AUDC2	Tonkanal 2 Kontrollregister	D203	53763			
AUDC3	Tonkanal 3 Kontrollregister	D205	53765			
AUDC4	Tonkanal 4 Kontrollregister	D207	53767			
AUDCTL	Sound Kontrollregister	D208	53768			
AUDF1	Frequenz von Tonkanal 1	D200	53760			
AUDF2	Frequenz von Tonkanal 2	D202	53762			
AUDF3	Frequenz von Tonkanal 3	D204	53764			
AUDF4	Frequenz von Tonkanal 4	D206	53766			
CHACTL	Zeichensatz Kontrollregister	D401	54273	CHART	2F3	755
CHBASE	Zeichensatz Adreßregister	D409	54281	CHBASE	2F4	756
COLBK	Rahmenfarbe	D01A	53274	COLOR4	2C8	712
COLPF0	Zeichenfarbe 0	D016	53270	COLOR0	2C4	708
COLPF1	Zeichenfarbe 1	D017	53271	COLOR1	2C5	709
COLPF2	Zeichenfarbe 2	D018	53272	COLOR2	2C6	710
COLPF3	Zeichenfarbe 3	D019	53273	COLOR3	2C7	711
COLPM0	Farbe von Player-Missile 0	D012	53266	PCOLR0	2C0	704
COLPM1	Farbe von Player-Missile 1	D013	53267	PCOLR1	2C1	705
COLPM2	Farbe von Player-Missile 2	D014	53268	PCOLR2	2C2	706
COLPM3	Farbe von Player-Missile 3	D015	53269	PCOLR3	2C3	707
CONSOL	Register der Funktionstasten	D01F	53279			
DLISTH	höherwertiges Byte der Display-List	D403	54275	SDLSTH	231	561
DLISTL	niederwertiges Byte der Display-List	D402	54274	SDLSTL	230	560
DMACTL	DMA Kontrollregister	D400	54272	SDMCTL	22F	559
GRACTL	Grafik Kontrollregister	D01D	53277			
GRAFM	Grafik der Missiles	D011	53265			
GRAFP0	Grafik von Player 0	D00D	53261			
GRAFP1	Grafik von Player 1	D00E	53262			
GRAFP2	Grafik von Player 2	D00F	53263			
GRAFP3	Grafik von Player 3	D010	53264			
HITCLR	Lösche alle Kollisionsregister	D01E	53278			
HPOM0	Horizontale Position von Missile 0	D004	53252			
HPOM1	Horizontale Position von Missile 1	D005	53253			
HPOM2	Horizontale Position von Missile 2	D006	53254			
HPOM3	Horizontale Position von Missile 3	D007	53255			
HPOSP0	Horizontale Position von Player 0	D000	53248			
HPOSP1	Horizontale Position von Player 1	D001	53249			
HPOSP2	Horizontale Position von Player 2	D002	53250			
HPOSP3	Horizontale Position von Player 3	D003	53251			
HSCROL	Horizontales Feinscroll-Register	D404	54276			
IRQEN	IRQ Kontrollregister	D20E	53774	POKMSK	10	16
IRQST	IRQ Statusregister	D20E	53774			
KBCODE	Tastaturcode	D209	53769	CH	2FC	764
M0PF	Kollision Missile 0-Bildschirmgrafik	D000	53248			

Hardware-Register		Adresse		Schattenregister	Adresse	
Name	Beschreibung	Hex	Dez	Name	Hex	Dez
M1PF	Kollision Missile 1-Bildschirmgrafik	D001	53249			
M2PF	Kollision Missile 2-Bildschirmgrafik	D002	53250			
M3PF	Kollision Missile 3-Bildschirmgrafik	D003	53251			
M0PL	Kollision Missile 0-Player	D008	53256			
M1PL	Kollision Missile 1-Player	D009	53257			
M2PL	Kollision Missile 2-Player	D00A	53258			
M3PL	Kollision Missile 3-Player	D00B	53259			
NMIEN	NMI Kontrollregister	D40E	54286			
NMIRES	Löscht NMIST	D40F	54287			
NMIST	NMI Statusregister	D40F	54287			
P0PF	Kollision Player 0-Bildschirmgrafik	D004	53252			
P1PF	Kollision Player 1-Bildschirmgrafik	D005	53253			
P2PF	Kollision Player 2-Bildschirmgrafik	D006	53254			
P3PF	Kollision Player 3-Bildschirmgrafik	D007	53255			
P0PL	Kollision Player 0-Player	D00C	53260			
P1PL	Kollision Player 1-Player	D00D	53261			
P2PL	Kollision Player 2-Player	D00E	53262			
P3PL	Kollision Player 3-Player	D00F	53263			
PACTL	Port A Kontrollregister	D302	54018			
PAL	Fernsehnorm	D014	53268			
PBCTL	Port B Kontrollregister	D303	54019			
PENH	Horizontale Position des Lichtgriffels		54284	LPENH	234	564
PENV	Vertikale Position des Lichtgriffels		54285	LPENV	235	565
PMBASE	Adresse des Player-Missile Bereichs		54279			
PORTA	Adresse von Port A	D300	54016			
PORTB	Adresse von Port B	D301	54017			
POT0	Drehregler 0	D200	53760	PADDL0	270	624
POT1	Drehregler 1	D201	53761	PADDL1	271	625
POT2	Drehregler 2	D202	53762	PADDL2	272	626
POT3	Drehregler 3	D203	53763	PADDL3	273	629
POT4	Drehregler 4	D204	53764	PADDL4	274	630
POT5	Drehregler 5	D205	53765	PADDL5	275	631
POT6	Drehregler 6	D206	53766	PADDL6	276	632
POT7	Drehregler 7	D207	53767	PADDL7	277	633
POTGO	Starte Drehregler Abfrage	D20B	53771			
PRIOR	Prioritätsregister	D01B	53275	GPRIOR	26F	623
RANDOM	Zufallszahlengenerator	D20A	53770			
SERIN	Serieller Eingabe-Port	D20E	53774			
SEROUT	Serieller Port Ausgaberegister	D20D	53773			
SIZEM	Größe der Missiles	D00C	53260			
SIZEP0	Größe Player 0	D008	53256			
SIZEP1	Größe Player 1	D009	53257			
SIZEP2	Größe Player 2	D00A	53258			
SIZEP3	Größe Player 3	D00B	53259			
SKCTL	Serieller Port Kontrollregister	D20F		SSKCTL	232	562
SKREST	Löscht SKSTAT	D20A	53770			
SKSTAT	Serieller Port Status	D20F	53775			
STIMER	Starte Timer	D209	53769			
TRIG0	Feuerknopf Joystick 0	D010	53264	STRIG0	284	644
TRIG1	Feuerknopf Joystick 1	D011	53265	STRIG1	285	645
TRIG2	Feuerknopf Joystick 2	D012	53266	STRIG2	286	646
TRIG3	Feuerknopf Joystick 3	D013	53267	STRIG3	285	647
VCOUNT	Vertikaler Bildschirmzeilenzähler	D40B	54283			
VDELAY	Verschiebt Player-Missiles eine Zeile	D01C	54276			
VSCROL	Vertikales Feinscrollregister	D405	54277			
WSYNC	Warte auf vertikales Synchronsignal	D40A	54282			

Grafikzauberei

In die Atari-Computer ist ein spezieller Mikroprozssor eingebaut, der die gesamte Grafiksteuerung des Computers übernimmt. Mit ein paar Kenntnissen über die Funktionsweise des Bausteins lassen sich wahre Grafikwunder auf dem Bildschirm erzeugen.

Als Atari die 8-Bit-Computer entwickelte, wollte man ein möglichst leistungsstarkes Gerät auf den Markt bringen. Zum damaligen Zeitpunkt war dies nur durch Hardware möglich, die den Mikroprozessor des Computers soweit wie möglich entlastet. Also ersannen die Entwickler des Computers drei spezielle Chips. Zwei dieser Bausteine haben die Aufgabe, den Ton im Computer zu erzeugen, die Tastatur abzufragen und mit angeschlossenen Peripheriegeräten in Verbindung zu treten. Der dritte Chip ist speziell für die Grafik verantwortlich und trägt den Namen »ANTIC«, »**A**lpha **N**umeric **T**elevision **I**nterface **C**hip«.

Dieser Baustein ist ein kompletter kleiner Mikroprozessor mit einer eigenen Programmiersprache. Die Befehle dieser Programmiersprache bestimmen, welche Grafik man auf dem Bildschirm sieht. Ein in dieser Programmiersprache geschriebenes Programm nennt man »Display-List«. Die Display-List steht dabei an beliebiger Stelle im Computerspeicher, wobei dem ANTIC vom 6502-Prozessor mitgeteilt wird, wo sich die Display-List befindet. Der ANTIC liest dann genau wie ein richtiger Mikroprozessor Befehl auf Befehl aus dem Speicher. Jeder einzelne dieser Befehle

Ein Prozessor namens ANTIC...

stellt eine bestimmte Grafik-Betriebsart dar. Wenn Sie den Computer zum Beispiel einschalten, haben Sie den ganz normalen blauen Bildschirm vor sich. Damit lassen sich 24 Zeilen ausgeben. Für jede dieser 24 Zeilen steht ein Befehl in der Display-List, der besagt, daß jetzt eine Zeile mit 40 normalen Zeichen dargestellt werden soll. In der Grafikstufe 2 lassen sich dagegen zehn Zeilen mit doppelt so hohen Zeichen sowie vier Zeilen mit normalen Zeichen abbilden. Die Display-List besteht hier aus zehn Befehlen für die Zeichen mit doppelter Höhe und vier Befehlen für die normalen Zeichen. Jedesmal, wenn Sie unter Basic einen neuen Grafik-Befehl eintippen, wird dem ANTIC einfach nur eine neue Display-List übergeben.

Es existieren drei verschiedene Arten von Befehlen in der Display-List. Die erste Art sagt dem ANTIC, daß er jetzt eine bestimmte Grafikstufe generieren muß. In dieser Art sind alle Befehle enthalten, die auf dem Bildschirm eine Grafik in einer wie auch immer gearteten Form erzeugen. Je nach Grafikbefehl für den ANTIC weiß dieser auch gleichzeitig, wie viele Bytes er aus dem Speicher holen muß und wie er diese Bytes zu interpretieren hat. Zum Beispiel muß er bei der Grafikstufe 8 insgesamt 40 Byte aus dem Speicher lesen. Die einzelnen Bits der Bytes verwendet er dann dazu, auf dem Bildschirm ein Punktemuster zu erzeugen, jedes gesetzte Bit läßt einen Punkt aufleuchten. Stößt er dagegen auf einen Befehl, der ein Zeichen auf dem Bildschirm darstellen soll, so holt er sich zunächst auch 40 Byte aus dem Speicher. Jedes dieser Bytes sagt dem ANTIC die Nummer des Zeichens, für das das Byte im Speicher steht und das auf dem Bildschirm dargestellt werden soll. Darauf hin liest er 8 weitere Bytes und interpretiert die Bits dieser Bytes als Punktemuster. Auf diese Weise wird ein Zeichen auf dem Bildschirm dargestellt.

...ein »Display-List«-Programm

Von diesen Grafikbefehlen gibt es 16 verschiedene Arten. Mit den letzten 15 lassen sich die verschiedensten Grafikstufen auf dem Bildschirm erzeugen. Die erste ist jedoch einzig und allein dazu da, Leerzeilen auf den Bildschirm zu bringen. Im deutschen PAL-Fernsehsystem stehen 625 Fernsehzeilen zur Verfügung. Da aber jedes zweite Bild praktisch dem ersten entspricht, halbiert sich diese Zeilenzahl auf 312,5. Würde der ANTIC nun gleich in der ersten Zeile mit dem Bildschirmaufbau beginnen, so würde man gut und gerne das oberste Viertel des Bildes auf dem Bildschirm nicht sehen. Vielleicht kennen Sie den Effekt, wenn Sie noch einen alten Fernseher besitzen, der nicht korrekt eingestellt ist, und bei dem das Bild ein wenig zu tief sitzt. Dort sehen Sie dann einen schwarzen Balken, in dem einige Zeilen wild flimmern. Dieser Balken sollte normalerweise nicht sichtbar sein, da er so weit nach oben verschoben ist, daß er nicht mehr das Bild verunziert. Worauf es aber ankommt: Dieser Balken besteht aus den ersten Fernsehzeilen. Wenn der ANTIC also gleich in der ersten Fernsehzeile mit dem Bildschirmaufbau beginnen würde, wäre der Anfang des Bildes dort, wo normalerweise dieser schwarze Balken sitzt, nämlich außerhalb des Bildschirms. Aus diesem Grund gibt es für den ANTIC Befehle, die nichts weiter verrichten, als den Bildschirm mit schwarzen Zeilen zu füllen. Dadurch läßt sich auf einfache Weise softwaremäßig ein Bild nach oben oder unten verschieben, man muß nur die entsprechende Anzahl an Leerzeilen einfügen oder entfernen. Von diesen sogenannten »Blank-Line-Befehlen« gibt es insgesamt acht Stück, sie erzeugen eine bis acht leere Zeilen auf dem Bildschirm. Eine normale Display-List besitzt am Anfang immer drei Befehle, die jeweils acht Blank-Lines erzeugen, insgesamt sind das 24 Leerzeilen.

Die zweite Befehlsgruppe umfaßt die Befehle, die dem ANTIC mitteilen, von woher er die Daten holen soll, aus denen er die Bildschirmgrafik aufbaut. Eigentlich ist dies gar keine eigene Befehlsgruppe, denn im ANTIC sind diese Befehle so gelöst, daß ein Grafikbefehl auch gleichzeitig ein Befehl zum Adressieren der Bildschirmgrafik sein kann. Der Unterschied liegt darin, daß in dem Grafikbefehl ein weiteres Bit gesetzt ist, das dem ANTIC mitteilt, daß jetzt ein Befehl zum Bestimmen der Adresse der Bildschirmgrafik kommt. Außerdem besteht solch ein

Systemadressen für Display-Lists			
DLISTL	54274	D402	Low-Byte Startadresse Display-List
DLISTH	54275	D403	High-Byte Startadresse Display-List
DMACTL	54272	D400	schaltet ANTIC ein und aus
SDLSTL	560	230	Schattenregister von DLISTL
SDLSTH	561	231	Schattenregister von DLISTH
SDMCTL	559	22F	Schattenregister von DMACTL

Befehl nicht aus einem, sondern aus drei Byte, dem Byte für die Grafikstufe, sowie zwei Byte, die die Adresse bestimmen, ab der der Computer die Byte für die Bildschirmgrafik holt. Dieser Befehl hat noch eine weitere Funktion. Aufgrund der inneren Konstruktion des ANTIC kann ein zusammenhängender Bildschirmspeicher maximal nur vier KByte lang sein. Es gibt aber Grafikstufen, die wesentlich mehr als vier KByte Bildschirmspeicher verbrauchen. Grafikstufe 8 benötigt zum Beispiel einen Bildschirmspeicher von über sieben KByte. In solchen Fällen wird in der Display-List einfach immer dann ein neuer Speicherbeginn angegeben, wenn der ANTIC die 4-KByte-Grenze überschreitet. Übrigens wird dieser Befehl auch benutzt, um die Bildschirme in unserem 256-Farben-Programm umzuschalten. Bei diesem Programm wird alle 50stel Sekunde zwischen zwei Bildschirmen hin- und hergeschaltet. Dies geschieht, indem eine VBLANK-Routine periodisch in die Display-List die beiden Startadressen der beiden Speicherbereiche schreibt.

Die dritte Befehlsgruppe der Display-List-Befehle umfaßt die Sprungbefehle. Ähnlich wie beim maximalen Bildschirmspeicher darf die Display-List nicht länger als 1 KByte sein. Da es vorkommt, daß sie doch mal länger wird, haben die Entwickler des ANTIC den Sprungbefehl eingebaut. Und wie bei jedem guten Mikroprozessor gibt es auch beim ANTIC bedingte und unbedingte Sprünge. Die unbedingten Sprungbefehlte verknüpfen einfach mehrere Display-Lists miteinander. Diese lassen sich also völlig wirr im Speicher verteilen. Sobald der ANTIC auf einen derartigen Befehl stößt, springt er sofort zu der Adresse, die hinter dem Befehl angegeben ist. Deshalb ist der unbedingte Sprungbefehl auch 3 Byte lang. Der bedingte Sprungbefehl übernimmt eine ganz spezielle Aufgabe im ANTIC. Was würde passieren, wenn der ANTIC einmal eine Display-List durchgearbeitet hat? Er würde versuchen, die Daten, die hinter der Display-List ste-

...ein paar Befehle...

hen, als Grafik-Befehle zu interpretieren. Und nachdem er 1 KByte durchgeackert hat, würde er von vorne die Display-List abarbeiten. Nur leider würde das nicht mehr synchron zum Bildschirm erfolgen. Aus diesem Grund gibt es den bedingten Sprungbefehl. Man benützt ihn, um am Ende einer Display-List wieder an dessen Anfang zu springen. Stößt der ANTIC auf dieses Kommando, verzweigt er nicht gleich zu der Adresse hinter dem Sprungbefehl, sondern wartet erst einmal ab, bis der Fernseher oder Monitor ein neues Bild zeichnet. Erst dann springt er zu der neuen Adresse, die in den meisten Fällen wieder der Anfang der Display-List ist.

Damit wären die drei Grundbefehlsarten der Display-List beschrieben. Jeder Grafik-Befehl übernimmt außerdem bis zu drei zusätzliche Funktionen, das hängt davon ab, welches Bit im Grafikbefehl noch gesetzt ist. Zwei Bit im Grafik-Befehl sind dafür reserviert, horizontales und vertikales

...und fertig ist die Grafik

Feinscrolling zu realisieren. Sind diese Bits gesetzt, wird eine Grafikzeile um genau so viele Pixel nach links verschoben, wie in dem jeweiligen Feinscrollingregister festgesetzt sind. Indem man diese Register verändert, läßt sich auf einfache Weise Feinscrolling verwirklichen.

Die letzte Möglichkeit betrifft das Bit, das einen Display-List-Interrupt auslöst. Da die Möglichkeiten des Display-List-Interrupts sehr vielfältig sind, ist diesem Thema ein eigener Beitrag (»Als die Farben laufen lernten«) gewidmet. Alle Display-List-Befehle sind in der Tabelle zu diesem Beitrag zusammengefaßt.

Wie muß man nun vorgehen, um eine eigene Display-List zu verwenden? Zunächst muß feststehen, an welcher Stelle die Display-List im Speicher steht. Diese Adresse schreibt man in die beiden Register »SDLSTL« (Adresse 560, $230) und »SDLSTH« (Adresse 561, $231), beziehungsweise in die Hardware-Register, wenn Sie die VBLANK-Routine nicht benutzen (»DLISTL«, Adresse: 54274, $D402 und »DLISTH«, Adresse: 54275, $D403). Vorher sollte man jedoch den ANTIC über die Speicherstelle »SDMCTL« (Adresse: 559, $22F) beziehungsweise über das Hardware-Register »DMACTL« (Adresse: 54272, $D400) abgeschaltet haben, da es sonst manchmal vorkommt, daß das Bild häßlich zu flackern anfängt. Sobald der vertikale Zeilenzähler »VCOUNT« (Adresse: 54283, $D40B) auf Null steht und damit anzeigt, daß der Elektronenstrahl des Fernsehers oder Monitors wieder in der obersten Zeile angelangt ist, sollte man den Bildschirm wieder einschalten.

Damit wäre alles Wissenswerte über die Display-List beschrieben. Sie sollten ausgiebig damit experimentieren, damit Sie ein Gefühl für die Grafikfähigkeiten des Computers bekommen. (hf)

ANTIC-Befehle																
Horizontales Scrolling		xx		xx		xx		xx		xx		xx		xx		xx
Vertikales Scrolling			xx	xx			xx	xx			xx	xx			xx	xx
Bildadresse festlegen					xx	xx	xx	xx					xx	xx	xx	xx
Display-List-Interrupt									xx	xx	xx	xx	xx	xx	xx	xx
1 Leerzeile	00								80							
2 Leerzeilen	10								90							
3 Leerzeilen	20								A0							
4 Leerzeilen	30								B0							
5 Leerzeilen	40								C0							
6 Leerzeilen	50								D0							
7 Leerzeilen	60								E0							
8 Leerzeilen	70								F0							
Sprunganweisung	01								81							
Springe und warte auf neues Bild	41								C1							
Normaler Zeichensatz (GR. 0)	02	12	22	32	42	52	62	72	82	92	A2	B2	C2	D2	E2	F2
Zeichensatz mit Unterlänge	03	13	23	33	43	53	63	73	83	93	A3	B3	C3	D3	E3	F3
Farbzeichensatz (GR. 12)	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
Farbzeichensatz (GR. 13)	05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
Doppelt breite Zeichen (GR. 1)	06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
Doppelt breite Zeichen (GR. 2)	07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
Grafik (GR. 3)	08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
Grafik (GR. 4)	09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
Grafik (GR. 5)	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
Grafik (GR. 6)	0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
Grafik (GR. 14)	0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
Grafik (GR. 7)	0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
Grafik (GR. 15)	0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE	BE	CE	DE	EE	FE
Grafik (GR. 8)	0F	1F	2F	3F	4F	5F	6F	7F	8F	9F	AF	BF	CF	DF	EF	FF

Als die Farben

Obwohl der Atari nur maximal fünf Farben auf dem Bildschirm darstellt, strotzen einige Programme nur so vor Farben. Die Erklärung liegt bei einem einfachen aber wirkungsvollen Interrupt des Computers.

Von Atari gibt es sehr schöne Grafik-Demonstrationsprogramme. Erinnern Sie sich an den grünen Roboter auf dem Fließband, der dem Betrachter scheinbar entgegenmarschiert? Oder an das in dem gleichen Demonstrationsprogramm gezeigte Raumschiff, das zu heroischer Musik durch den Weltraum rast und dabei im Takt der Musik auf und ab schwebt? Sehr schön ist auch ein Programm, bei dem sich das Atari-Zeichen um seine Mittelachse dreht und im Vordergrund ein Pelikan quer über den Bildschirm flattert. Eins haben alle diese Programme gemeinsam: Bei allen ist das Atari-Logo an irgendeiner Stelle der Demonstration zu sehen. Und in dem Zeichen laufen regenbogenartig Farben nach oben. Dabei sind auf dem Bildschirm je nach Größe des Atari-Zeichens zwischen 50 und 100 Farben zu sehen. Um solche Effekte zu erzeugen, bedienen sich die Programmierer der Diplay-List-Interrupts des Computers.

Was sind nun Display-List-Interrupts? Kurz gesagt, damit wird bei einer bestimmten Bildschirmzeile im Computer ein Interrupt ausgelöst. Der Mikroprozessor springt dann ein spezielles Programm an, mit dem zum Beispiel die Bildschirmfarbe umgeschaltet wird. Die Bildschirmzeile, in der der Interrupt auftreten soll, sowie das Programm, das vom Mikroprozessor angesprungen wird, ist vom Programmierer frei definierbar. Üblicherweise besteht das Programm aus wenigen Maschinensprache-Befehlen, mit denen die Farbe am Bildschirm umgeschaltet wird. Der Display-List-Interrupt ist jedoch für beliebige andere Anwendungen einsetzbar. Zum Beispiel lassen sich damit ab einer bestimmten Bildschirmzeile die Position sowie der Speicherbereich für die Player-Missiles neu festlegen. Auf diese Weise bekommt man statt der üblichen vier Player und Missiles jeweils acht oder mehr Objekte auf den Bildschirm.

Denkbar wäre auch, daß mit dem Display-List-Interrupt eine Musik gespielt wird. Musikstücke, die der Computer abspielt, während er gleichzeitig noch mit etwas anderem beschäftigt ist, werden normalerweise mit dem VBLANK-Interrupt realisiert (siehe Beitrag »Das Programm im Hintergrund« in diesem Heft). Der VBLANK-Interrupt wird alle 50stel Sekunde ausgeführt, wodurch im Takt dieser Zeitspanne ein neuer Ton beginnt. Definiert man nun einen Display-List-Interrupt, der in der Mitte des Bildschirms ausgelöst wird und der die gleiche Musikroutine wie der VBLANK-Interrupt verwendet, so steht für das Musikprogramm die doppelte Notenauflösung zur Verfügung. Das Musikprogramm wird in dem Fall doppelt so oft aufgerufen. Dies sind nur einige anregende Beispiele für den Einsatz des Display-List-Interrupts. Das Spektrum der Anwendungen ist schier endlos, und so manches Spiel oder Anwenderprogramm hat schon davon profitiert (siehe »Ballblazer« von Lucasfilm Games oder »StarTexter« von Sybex).

Stolpersteine in der Display-List

Doch wie wird ein Display-List-Interrupt programmiert? Zunächst muß festgelegt werden, in welcher Bildschirmzeile der Interrupt ausgelöst werden soll. Dies wird über die Display-List festgelegt. Jeder der Display-List-Befehle ist so programmierbar, daß er einen Display-List-Interrupt auslöst. Dafür ist in jedem Befehl ein Bit reserviert, das bei gesetztem Zustand einen Interrupt erzeugt (siehe Beitrag »Grafikzauberei« in diesem Heft). Zu beachten ist dabei, daß der Interrupt erst ausgelöst wird, wenn der Display-List-Befehl bereits abgearbeitet ist. Er tritt also immer in der letzten Bildschirmzeile eines Befehls auf. Aus diesem Grund muß der Interrupt-Befehl in der Display-List immer vor dem Befehl stehen, in dem der Interrupt ausgewertet wird. Zudem braucht der Prozessor einige Zeit, um zu erkennen, daß ein Display-List-Interrupt auftrat. Bis dann vom Prozessor das eigentliche Programm angesprungen wird, ist der Elektronenstrahl auf dem Bildschirm fast bei der nächsten Bildschirmzeile angelangt.

Das Programm, das durch den Display-List-Interrupt aufgerufen wird, muß aus Zeitgründen in Maschinensprache programmiert werden. Wichtig bei einer Display-List-Routine ist die Tatsache, daß beim Aufruf der Routine alle Prozessor-Register noch erhalten sind. Die in dem Programm verwendeten Register müssen also erst gesichert werden, zum Beispiel auf dem Stack. Nehmen wir an, daß wir eine Display-List-Interrupt-Routine programmieren wollen, die an einer Stelle des Bildschirms die Farben umschaltet. Dazu wird höchstens ein Register benötigt, in diesem Fall der Accumulator. Der erste Befehl der Display-List-Interrupt-Routine muß also

```
PHA
```

lauten. Damit wird der Inhalt des Accus auf dem Stack gesichert. Als nächstes wird der Accu mit dem Farbwert geladen, auf den umgeschaltet werden soll, und das Farbregister mit diesem Wert geladen.

```
LDA #$30 ;Farbe Rot
STA $D018 ;als Hintergrundfarbe
```

Danach muß der Accu wieder mit dem Wert geladen werden, den er bei Eintritt in die Interrupt-Routine enthielt. Die Routine wird anschließend wieder verlassen, und der Mikroprozessor nimmt seine unterbrochene Arbeit wieder auf.

```
PLA
RTI
```

Das war schon die ganze Display-List-Routine zum Umschalten der Hintergrundfarbe. Je nach Aufgabenstellung muß die Routine entsprechend erweitert werden. Elementar wichtig ist jedoch, daß das Programm alle Register, die in der Routine benutzt werden, auf dem Stack sichert, und daß es diese vor dem Verlassen der Routine wieder restauriert.

Nachdem feststeht, wo ein Interrupt auftreten soll und die Routine fertig ist, muß dem Computer mitgeteilt werden, ab welcher Adresse die Routine im Computerspeicher steht. Dies geschieht über »VDSLST« (Adressen 512 und 513, $200 und $201). In VDSLST wird dabei das Low-Byte der Adresse der Interrupt-Routine geschrieben und in VDSLST+1 das High-Byte. Tritt nun ein Interrupt auf, springt der Prozessor über diese Adresse in die Interrupt-Routine. Mit der Speicherstelle »NMIEN« (Adresse 54286, $D40E) wird der Interrupt eingeschaltet, indem Bit 7 dieser Adresse gesetzt wird (siehe auch Seite 125). Abgeschaltet wird die Routine wieder, indem dieses Bit gelöscht, oder wenn die <RESET>-Taste auf der Tastatur gedrückt wird.

In Verbindung mit den Display-List-Interrupts spielt das Register

laufen lernten

»WSYNC« (Adresse 54282, $D40A) eine wichtige Rolle. Wenn Sie das obige Assembler-Programm eingegeben haben und den Interrupt wie erklärt einschalten, werden Sie auf dem Bildschirm einen Farbwechsel sehen.

Der Übergangsbereich der Farben vor und nach dem Interrupt flackert jedoch. Der Interrupt wird immer am Anfang der letzten Zeile eines Display-List-Befehls ausgelöst. Bis der Mikroprozessor jedoch die Interrupt-Routine angesprungen hat, ist der Elektronenstrahl auf dem Bildschirm ein gutes Stück weitergewandert. Nachdem der Strahl dann gut drei Viertel der Bildschirmzeile beschrieben hat, stößt der Mikroprozessor auf den »STA $D018«-Befehl, womit die Bildschirmfarbe schlagartig umgeschaltet wird. Auf dem Bildschirm ist dadurch eine unansehliche Treppe zu erkennen, die obendrein noch zittert. Zur Vermeidung dieses Effekts ist die Speicherstelle WSYNC vorhanden.

Mit sieben Befehlen zum Regenbogen

Schreibt der Mikroprozessor irgendeinen Wert in diese Speicherstelle, so wird er bis zum Anfang der nächsten Bildschirmzeile angehalten. Dies läßt sich ausnutzen, um die häßliche Treppe auf dem Bildschirm zu vermeiden. Folgendes Programm soll dies verdeutlichen:

```
PHA
LDA #$30
STA WSYNC
STA $D018
PLA
RTI
```

Jetzt wartet der Mikroprozessor mit dem Umschalten der Farbe bis zum Beginn einer neuen Zeile. Damit ist der Farbübergang hinter den nicht sichtbaren linken Rand des Bildschirms gerückt.

Mit der Speicherstelle WSYNC wird auch der Farblauf in den verschiedenen anfangs angesprochenen Atari-Grafikdemonstrationen programmiert. Diese legen eine Speicherstelle fest, die nach jedem Interrupt um 1 erhöht wird, sollen die Farben nach oben laufen, oder um 1 vermindert wird, wenn sie nach unten laufen. Anschließend lädt der Accu diese Speicherstelle. Dann erfolgt der obligatorische Schreibbefehl in WSYNC und das Umschalten der Farben. Der Accu wird nun um 1 erhöht oder vermindert, je nachdem, auf welche Weise der Farblauf auf dem Bildschirm erscheinen soll (von Hell nach Dunkel oder umgekehrt). Anschließend wird dieser Wert wieder in WSYNC geschrieben und die Prozedur so lange wiederholt, bis der Bereich auf dem Bildschirm, in dem die Farben laufen, abgedeckt ist.

Ohne Probleme lassen sich auch mehrere Interrupts hintereinander programmieren. Zum Beispiel soll der Bildschirm in drei Farben aufgeteilt werden, der obere Bereich Rot, in der Mitte Grün und unten Blau. Zunächst setzt man die Bildschirmfarbe auf Rot. Dann bestimmt man in der Display-List, an welcher Stelle auf dem Bildschirm die Farben umgeschaltet werden sollen. Gleichzeitig müssen zwei Interrupt-Routinen programmiert werden. Die eine schaltet die Bildschirmfarbe auf Grün und die zweite auf Blau. Die erste setzt aber, nachdem sie die Farbe umgeschaltet hat, die Startadresse der Display-List-Interrupt-Routine auf die zweite Routine. Die zweite führt das Entsprechende mit der ersten Routine durch. Auf diese Art ist gewährleistet, daß die Routinen immer hintereinander aufgerufen werden. Der Bildschirm wird durch die VBLANK-Routine automatisch auf Rot geschaltet. Schon haben Sie drei Farben untereinander auf dem Bildschirm.

Damit wär auch schon alles über die Programmierung von Display-List-Interrupts gesagt. Mit wenigen Befehlen lassen sich erstaunliche Effekte erzielen. Mit Ihrem neu erworbenen Wissen dürfte es Ihnen keine Probleme bereiten, zu verstehen, wie zum Beispiel »Ballblazer« programmiert wurde. Beim Vor- und Zurückfahren werden einfach andere Display-Lists mit anderen Interruptstellen verwendet. Fertig ist der Fahreffekt. Mit Display-List-Interrupts lassen sich auch Anwenderprogramme optisch enorm aufmöbeln (siehe »ASS« in diesem Heft). Und so schwierig ist das gar nicht. Probieren Sie es aus. (hf)

Systemadressen für Display-List-Interrupts

VDSLST	512	200	Low-Byte der Adresse der Interrupt-Routine
	513	201	High-Byte der Adresse der Interrupt-Routine
NMIEN	54286	D40E	Bit 7 schaltet DLIs ein und aus
WSYNC	54282	D40A	läßt den Prozessor auf den Beginn der nächsten Bildschirmzeile warten

Inserentenverzeichnis

Die Handschrift

Als Programmierer ist man nicht ausschließlich auf den eingebauten Zeichensatz der XL/XE-Computer angewiesen. Neue Zeichensätze sind ohne Probleme nachrüstbar.

Kennen Sie »Turbo-Basic«? Dann ist Ihnen auch das Happy-Computer-Zeichen bekannt, das am oberen Rand des Bildschirms steht, während Turbo-Basic lädt. Um das Zeichen auf dem Bildschirm darzustellen, wurde eine spezielle Eigenschaft der Atari-Computer ausgenutzt: Es lassen sich beliebige neue Zeichensätze definieren. Ob man nun Umlaute und das deutsche »ß« zur Verfügung haben möchte, ob man ein Copyright-Zeichen benötigt oder einfach nur in kyrillischer Schrift programmieren möchte, mit umdefinierten Zeichensätzen ist das alles machbar.

Ein Zeichen wird auf dem Bildschirm in einer Acht-mal-acht-Punktmatrix dargestellt. Jeweils acht nebeneinanderliegende Punkte werden in einem Byte zusammengefaßt. Für ein komplettes Zeichen benötigt man also acht Byte. Diese werden im Speicher hintereinander abgelegt, wobei die erste Zeile eines Zeichens das erste Byte ist, die zweite Zeile das zweite Byte und so weiter.

Wenn Sie ein Zeichen entwerfen, müssen Sie eine Besonderheit des Atari-Computers beachten: In einem Zeichen müssen horizontal immer zwei Punkte nebeneinander liegen, da der Punkt sonst kaum sichtbar ist. Wahrscheinlich kennen Sie den Effekt, wenn Sie in Grafikstufe 8 eine vertikale Linie ziehen. Schalten Sie Ihren Computer ein und geben Sie im Basic folgende Befehlsfolge ein:

```
GRAPHICS 8
SETCOLOR 2,0,0
COLOR 1
PLOT 160,10:DRAWTO 160,150
```

Sie sehen in der Mitte des Bildschirms jetzt eine blasse vertikale Linie. Geben Sie nun zusätzliche, folgende Zeile ein:

```
PLOT 161,10:DRAWTO 161,150
```

Sofort erstrahlt die Linie in hellem Weiß. Die gleichen Effekte würden auch in Zeichensätzen auftreten, denn die Punkteauflösung auf dem Bildschirm ist in Grafikstufe 0 und 8 gleich (ein Zeichen besteht vertikal aus acht Punkten, und 40 Zeichen lassen sich nebeneinander darstellen, macht also 320 Punkte wie in Grafikstufe 8).

Dieses merkwürdige Verhalten hängt mit dem Videosignal des Computers zusammen. Da sich das Gerät technisch seit seinem Erscheinen im Jahr 1979 nicht verändert hat, ist immer noch das gleiche Videoteil eingebaut. Wahrscheinlich wußte Atari zum damaligen Zeitpunkt nicht, wie mit vertretbarem Aufwand und vor allem in vertretbarem finanziellen Rahmen ein besseres Videoteil eingebaut werden könnte.

Ganz nebenbei lassen sich dadurch bei einem Farbfernsehapparat oder Monitor hübsche Effekte erzeugen. Geben Sie dazu folgendes Programm ein:

```
100 GRAPHICS 8
110 SETCOLOR 2,0,0:COLOR 1
120 FOR I=0 TO 319 STEP 2
130 PLOT I,0:DRAWTO I,159
140 NEXT I
```

Wenn Sie das Programm laufen lassen, sehen Sie, wie sich der Bildschirm langsam mit dunkelgrünen und hellblauen vertikalen Linien überzieht. Bei einem amerikanischen System mit der NTSC-Fernsehnorm würden diese in sattem Purpur und Blau leuchten. Einige amerikanische Spiele nutzen diesen Effekt gebührend aus. Das führt dann dazu, daß zum Beispiel »Ultima III« in Amerika vor Farben strotzt, in Deutschland dagegen ist das Spiel trostlos grau in grau.

Doch zurück zu den Zeichensätzen. Je nach gewählter Grafikstufe lassen sich gleichzeitig unterschiedlich viele Zeichen darstellen. In Grafikstufe 0 sind es 128 Zeichen, in Grafikstufe 1 und 2 dagegen nur 64. Damit nimmt ein Zeichensatz je nach gewählter Grafikstufe auch verschieden viel Speicher in Anspruch. In Grafikstufe 0 benötigt er 1 KByte, in Grafikstufe 1 und 2 nur ein halbes KByte.

Der Zeichensatz bildet im Speicher des Computers einen zusammenhängenden Block von Bytes. Nachdem man einen Zeichensatz zusammengestellt hat und dieser im Speicher steht, muß man dem Computer nur noch mitteilen, an welcher Adresse sich der neue Zeichensatz befindet. Beim Atari-Computer gibt es nun die Einschränkung, daß der Zeichensatz nicht an jeder beliebigen Stelle im Speicher stehen darf. Vielmehr muß er bei Grafikstufe 0 an der Grenze zum nächsten vollen KByte im Computer stehen, also zum Beispiel bei Adresse 16384, 17408 oder 18432 ($4000, $4400, $4800) und bei Grafikstufe 1 und 2 an der Grenze zum nächsten halben KByte, zum Beispiel bei Adresse 16384, 16896 oder 17408 ($4000, $4200, $4400). Aus diesem



Aussehen der Zeichensätze im Speicher und auf dem Bildschirm

des Computers

Grund reicht auch ein einziges Byte, um dem Computer den Beginn des Zeichensatzes mitzuteilen. Dies geschieht über die Speicherstelle 54281 ($D409), die den Namen »CHBASE« trägt. In diese Speicherstelle trägt man also das höherwertige Byte der Adresse des Zeichensatzes ein. Will man in Basic eigene Zeichensätze verwenden, so benutzt man besser das Schattenregister von CHBASE bei Adresse 756 ($2F4) mit dem Namen »CHBAS«. (Was es mit den Schattenregistern auf sich hat, lesen Sie in dem Beitrag »Programm im Hintergrund«.)

Damit Sie einen kompletten Zeichensatz programmieren können und Ihre Zeichen anschließend wiederfinden, müssen Sie noch wissen, in welcher Reihenfolge die Zeichen im Speicher stehen. Theoretisch müßte man annehmen, daß zum Beispiel das Zeichen »A« mit dem ASCII 65 auch die 65ste 8-Byte-Gruppe im Zeichensatz ist. Dem ist aber nicht so. Im Speicher stehen erst die Zeichen mit ASCII zwischen 32 und 95, dann folgen 32 Zeichen, die die Grafiksymbole enthalten, und am Schluß stehen 32 Zeichen mit den kleinen Buchstaben. Nur so lassen sich in Grafikstufe 1 und 2 Großbuchstaben, Zahlen und Sonderzeichen zusammen darstellen. Wäre der Zeichensatz im Speicher in der gleichen Reihenfolge wie der ASCII aufgebaut, so ließen sich Buchstaben und Sonderzeichen in diesen Grafikstufen nicht zusammen verwenden, denn diese Zeichen würden an zwei verschiedenen Halb-KByte-Grenzen liegen. Zur Verdeutlichung ein kleines Experiment. Geben Sie folgende Zeilen in den Computer ein:

```
GRAPHICS 1
PRINT #6;"HALLO, DU DA!"
```

Auf dem Bildschirm sehen Sie jetzt am oberen Rand eine orangefarbige Textzeile. Tippen Sie nun den Befehl

```
POKE 756,226
```

ein. Damit versetzen Sie den Zeichensatz um ein halbes KByte. Der Bildschirm ist jetzt übersät mit Herzchen, der Text in der ersten Zeile ist klein geschrieben, und das Komma und das Ausrufezeichen im Text haben sich in Grafikzeichen verwandelt. Wenn Sie jetzt folgende Zeilen eingeben,

```
GRAPHICS 0
PRINT "HALLO, DU DA!"
POKE 756,226
```

werden Sie anschließend keine Veränderung feststellen, denn in Grafikstufe 0 läßt sich der Zeichensatz nur in ganzen KByte-Schritten versetzen. Geben Sie aber

```
POKE 756,228
```

ein, um den Zeichensatz um ein komplettes KByte zu verschieben (der Originalzeichensatz liegt bei 224), so werden Sie auf dem Bildschirm nichts mehr erkennen können. Jetzt nimmt der Computer nämlich einen Speicherbereich als Zeichensatz an, dessen Bitmuster keine sinnvollen Symbole ergeben.

Nebenbei bemerkt gibt es in den XL/XE-Computern noch einen zweiten kompletten Zeichensatz, der anstelle der ohnehin selten benutzten Grafiksymbole zahlreiche Sonderzeichen enthält, darunter auch die deutschen Umlaute. Sie schalten diesen Zeichensatz ein, indem Sie den Befehl

```
POKE 756,204
```

eingeben. Leider hat dieser Zeichensatz nicht das deutsche »ß« eingebaut. Deshalb haben wir für Sie ein Programm abgedruckt, das den Originalzeichensatz vom ROM ins RAM kopiert und dann anstelle der Grafikzeichen die deutschen Umlaute und das »ß« einfügt.

Mit Zeichensätzen lassen sich noch einige Spielereien anstellen. Im Atari gibt es nämlich noch ein weiteres Register, das in Verbindung mit Zeichensätzen von Bedeutung ist. Wahrscheinlich haben Sie sich schon gewundert, daß ein Zeichensatz in Grafikstufe 0 aus nur 128 Zeichen besteht, obwohl in Basic mit dem CHR$-Kommando 256 Symbole ausgedruckt werden können. Die ersten 128 Zeichen werden dann normal dargestellt. Die zweiten 128 Zeichen entsprechen den ersten 128, mit dem Unterschied, daß sie invers ausgegeben werden (schwarze Schrift auf weißem Grund). Diese Zeichen sind nicht etwa invers programmiert, dafür wäre im Zeichensatz gar kein Platz vorhanden. Vielmehr stellt der Computer Zeichen mit einem ASCII größer als 127 automatisch invers dar. Dem Computer teilt man jedoch über das obengenannte Register mit, wie er die letzten 128 Zeichen darstellen soll. Dieses Register liegt bei Adresse 54273 ($D401) und trägt den Namen »CHACTL«. In Basic muß man jedoch dessen Schattenregister mit dem Namen »CHART« bei Adresse 755 ($2F3) benutzen.

Das Register übernimmt drei Funktionen. Setzt man Bit 0 in diesem Register, so werden die normalerweise invers dargestellten Zeichen unsichtbar. Geben Sie dazu folgende Zeilen ein:

```
GRAPHICS 0
FOR I=128 TO 255:? CHR$(I);:
NEXT I
```

Auf dem Bildschirm stehen jetzt sämtliche inverse Zeichen des Computers. Tippen Sie nun folgenden Befehl ein:

```
POKE 755,1
```

Alle inversen Zeichen sind vom Bildschirm verschwunden, auch der Cursor, der selbst ein inverses Leerzeichen ist. Wenn Sie nun anschließend

```
POKE 755,0
```

eingeben, sind alle Zeichen wieder normal dargestellt. Mit

```
POKE 755,2
```

haben Sie den Urzustand hergestellt, denn setzt man Bit 1, so werden die Zeichen mit ASCII größer 127 invers ausgegeben. Setzt man Bit 0 und Bit 1 – in Speicherstelle 755 muß also eine 3 stehen – werden alle normalerweise inversen Zeichen als weißer Block abgebildet. Bit 2 von CHART stellt alle Zeichen auf den Kopf. Über Sinn oder Unsinn dieser letzten Funktion läßt sich streiten. Auf jeden Fall hat sie der Computer eingebaut, und warum sollte man damit nicht mal den einen oder anderen Gag programmieren.

Neben den normalen Zeichensätzen kann der Atari farbige Zeichensätze auf den Bildschirm bringen. Sie kennen sicher die beiden Grafikstufen 12 und 13. Geben Sie einmal folgendes ein:

```
GRAPHICS 12
PRINT #6;"HALLO, DU DA!"
```

Auf dem Bildschirm sehen Sie jetzt in der oberen linken Ecke eine merkwürdige Ansammlung von Farbpunkten. Das liegt daran, daß der Computer in dieser Grafikstufe die einzelnen Bit eines Zeichens anders interpretiert. Das Zeichen ist jetzt nur vier Bild-

Systemadressen für Zeichensätze

CHACTL	54273	D401	Bit 0 – inverse Zeichen ausgefüllt Bit 1 – inverse Zeichen sichtbar Bit 2 – alle Zeichen auf dem Kopf
CHART	755	2F3	Schattenregister von CHACTL
CHBAS	756	2F4	Schattenregister von CHBASE
CHBASE	54281	D409	legt den Beginn eines Zeichensatzes fest
CHRORG	57344	E000	Beginn des eingebauten Zeichensatzes
CHRORU	52224	CC00	Beginn des Sonderzeichensatzes

punkte breit. Jeweils zwei nebeneinanderliegende Bit im Zeichensatz werden benutzt, um einen Bildschirmpunkt darzustellen. Mit zwei Bit lassen sich insgesamt vier verschiedene Kombinationen bilden. Jeder dieser Kombinationen wird ein entsprechendes Farbregister zugeordnet. »SETCOLOR 4,...« verändert die Farbe des Bildschirmpunktes mit dem Bitmuster »00«, »SETCOLOR 0,...« ist für »01«, »SETCOLOR 1,...« für »10« und »SETCOLOR 2,...« ist für »11« zuständig. Wie bei Grafikstufe 0 werden gleichzeitig maximal 128 Zeichen dargestellt. Und wie bei Grafikstufe 0 haben auch hier Zeichen über ASCII 127 eine spezielle Bedeutung. Druckt man diese Zeichen aus, so nehmen die Bildschirmpunkte mit dem Bitmuster »10« die Farbe an, die mit dem Befehl »SETCOLOR 3,...« festgelegt wird. Damit lassen sich in den Grafikstufen 12 und 13 fünf verschiedene Farben gleichzeitig auf dem Bildschirm darstellen.

Um diese Grafik-Betriebsart zu nutzen, muß man sich erst einen entsprechenden Zeichensatz zusammenbasteln. Ganz einfach läßt sich das ebenfalls mit dem in dieser Ausgabe abgedruckten »CHAR-MAKER« bewerkstelligen.

Zum Schluß gibt es noch eine Zeichensatzbetriebsart, die es erlaubt, Unterlängen in den einzelnen Zeichen darzustellen. Diese Betriebsart wird vom Betriebssystem des Computers nicht unterstützt. Deshalb muß man sich dafür eine eigene »Display-List« schreiben (wie das geht, wird im Beitrag »Grafikzauberei« beschrieben). Diese letzte Betriebsart stellt alle Zeichen auf dem Bildschirm zehn Zeilen hoch dar. Trotzdem werden nach wie vor im Zeichensatz nur acht Byte pro Zeichen verwendet. Der Trick bei der Sache ist, daß bei den letzten 32 Zeichen eines Zeichensatzes von einem Zeichen die ersten beiden Byte genommen und unter das Zeichen gesetzt werden. Die oberen beiden Zeilen eines Zeichens auf dem Bildschirm bleiben dann frei (siehe Seite 134). Auf diese Weise lassen sich ganz einfach Unterlängen darstellen. Voraussetzung dafür ist, daß die kleinen Buchstaben in den letzten 32 Stellen eines Zeichensatzes stehen, was beim Originalzeichensatz der Fall ist, und daß man sich einen Zeichensatz für Unterlängen umbastelt.

Damit wären sämtliche Wege zur Gestaltung von Zeichensätzen bei den Atari-XL-Computern beschrieben. Viel Spaß mit Ihren eigenen Kreationen und mit dem auf Seite 35 vorgestellten »CHAR-MAKER«!

(Andreas Sons/hf)

```
100 PRINT "{ESC CTL <}"                        <QM>
110 DIM TREE$(20):TREE$="!${ESC CTL +}{E
SC CTL +}{ESC CTL =}▲%{ESC CTL +}{ESC CT
L +}{ESC CTL =}#&{ESC CTL -}{ESC CTL -}{
ESC CTL -}":TREE$(6,6)=CHR$(34)                <FA>
120 POKE 106,PEEK(106)-4:GRAPHICS 0:DL=P
EEK(560)+PEEK(561)*256                         <WQ>
130 POKE DL+4,0:POKE DL+5,PEEK(106):FOR
A=0 TO 511:POKE PEEK(106)*256+A,PEEK(573
44+A):NEXT A                                   <UV>
140 POKE DL+5,PEEK(89):POKE DL+3,68:FOR
A=DL+6 TO DL+19:POKE A,4:NEXT A:POKE DL+
4,PEEK(88)                                     <RQ>
150 FOR A=PEEK(106)*256+8 TO PEEK(106)*2
56+55:READ B:POKE A,B:NEXT A                   <XO>
160 POKE 756,PEEK(106)                         <AG>
170 GOSUB 230                                  <SV>
180 POSITION 19,8:PRINT TREE$                  <IA>
190 POKE 710,36:POKE 712,0                     <WH>
200 POKE 708+INT(RND(0)*2),176+RND(0)*62       <AM>
210 FOR A=1 TO RND(0)*10:NEXT A                <ZO>
220 GOTO 200                                   <LV>
230 A=1536:RESTORE 360:POKE 54286,0            <OH>
240 READ B:IF B<256 THEN POKE A,B:A=A+1:
GOTO 240                                       <HD>
250 RESTORE 340:POKE 82,5:POKE 83,35           <SV>
260 POKE DL+20,134:POKE DL+21,12               <ZP>
270 POSITION 8,15:PRINT "tree":POKE 711,
148                                            <FW>
280 PRINT :PRINT "DIESES▲PROGRAMM▲SOLL▲D
IE▲MOEG-▲LICHKEIT▲ZUM▲UMDEFINIEREN▲DES"        <IV>
290 PRINT "ZEICHENSATZES▲DEMONSTRIEREN.▲
▲▲▲DER▲BAUM▲BESTEHET▲AUS▲:▲"                   <PK>
300 FOR A=33 TO 38:PRINT CHR$(A);:NEXT A
:PRINT "."                                     <QL>
310 POKE 82,0:POSITION 2,17:PRINT TREE$        <XV>
320 POKE 83,39:POSITION 36,17:PRINT TREE
$                                              <IB>
330 POKE 512,0:POKE 513,6:POKE 54286,192
:RETURN                                        <LF>
340 DATA▲0,72,48,12,15,233,54,7,179,35,3
,3,3,7,3,3,3,3,3,3,3,12,63,204                 <ZD>
350 DATA▲0,72,52,240,224,65,252,130,192,
192,192,192,192,192,224,192,192,192,192,
192,192,240,204,52                             <HX>
360 DATA▲72,169,0,141,10,212,141,24,208,
169,10,141,23,208,173,244,2,141,9,212,10
4,64,256                                       <WS>
```

Listing. »Tree« demonstriert anschaulich Farbzeichensätze beim XL/XE

```
Ok.

                00010 ;*************************
                00020 ;*                       *
                00030 ;* ZEICHENSATZUMWANDLUNG *
                00040 ;* auf deutsche Zeichen  *
                00050 ;*                       *
                00060 ;* 21.1.1987             *
                00070 ;*                       *
                00080 ;* Henrik Fisch          *
                00090 ;*                       *
                00100 ;*************************
                00110 ;
                00120 ; Dieses Programm setzt in den
                00130 ; Standart-Zeichensatz die
                00140 ; deutschen Umlaute ein.
                00150 ;
                00160 ;  1 = grosser A Umlaut
                00170 ;  9 = kleiner u Umlaut
                00180 ; 15 = grosser O Umlaut
                00190 ; 16 = kleiner o Umlaut
                00200 ; 19 = kleiner a Umlaut
                00210 ; 21 = grosser U Umlaut
                00220 ; 26 = deutsches sz
                00230 ;
                00240 ; VON DIESEM PROGRAMM BENUTZT
                00250 ;
00E0:           00260 GCHAR  .EQ $E0
00E2:           00270 SSTART .EQ $E2
00E4:           00280 SEND   .EQ $E4
00E6:           00290 DSTART .EQ $E6
00E8:           00300 ANZAHL .EQ $E8
                00310 ;
                00320 ; OS - EQUATES
                00330 ;
006A:           00340 RAMTOP .EQ $6A
                00350 ;
                00360 ; PROGRAMMSPEICHERUNG
                00370 ;
                00380  .OR $600
                00390  .TA $600
                00400  .TF "D:GCHAR.OBJ"
                00410 ;
                00420 ; PROGRAMMSTART
                00430 ;
0600: 68        00440  PLA
                00450 ;
                00460 ; ERMITTELT DEN SPEICHERBEREICH
                00470 ; FUER DEN ZEICHENSATZ
                00480 ;
                00490 START
0601: A5 6A     00500  LDA RAMTOP
0603: 85 E1     00510  STA GCHAR+1
0605: A9 00     00520  LDA #0
0607: 85 E0     00530  STA GCHAR
                00540 ;
                00550 ; KOPIERT DEN ZEICHENSATZ
                00560 ; VOM ROM INS RAM
                00570 ;
```

```
0609: A9 00    00580  LDA #0
060B: 85 E2    00590  STA SSTART
060D: A9 E0    00600  LDA #$E0
060F: 85 E3    00610  STA SSTART+1
0611: A9 00    00620  LDA #0
0613: 85 E4    00630  STA SEND
0615: A9 E4    00640  LDA #$E4
0617: 85 E5    00650  STA SEND+1
0619: A5 E0    00660  LDA GCHAR
061B: 85 E6    00670  STA DSTART
061D: A5 E1    00680  LDA GCHAR+1
061F: 85 E7    00690  STA DSTART+1
               00700 ;
0621: A0 00    00710  LDY #0
               00720 LOOP
0623: B1 E2    00730  LDA (SSTART),Y
0625: 91 E6    00740  STA (DSTART),Y
0627: C8       00750  INY
0628: D0 F9    00760  BNE LOOP
               00770 ;
062A: A5 E3    00780  LDA SSTART+1
062C: C5 E5    00790  CMP SEND+1
062E: F0 07    00800  BEQ LOOP1
0630: E6 E3    00810  INC SSTART+1
0632: E6 E7    00820  INC DSTART+1
0634: 4C 23 06 00830  JMP LOOP
               00840 LOOP1
               00850 ;
               00860 ; VERAENDERN DES ZEICHENSATZES
               00870 ;
0637: A9 00    00880  LDA #0
0639: 85 E8    00890  STA ANZAHL
               00900 ;
063B: A9 78    00910  LDA #CHARS     ;bestimmt die
063D: 85 E2    00920  STA SSTART     ;Adresse der zu
063F: A9 06    00930  LDA /CHARS     ;Ersetzendene
0641: 85 E3    00940  STA SSTART+1 ;Zeichen im RAM
               00950 ;
               00960 EXCHAN1
0643: A4 E8    00970  LDY ANZAHL
0645: B9 B8 06 00980  LDA CHRPOS,Y
0648: 0A       00990  ASL
0649: 0A       01000  ASL
064A: 0A       01010  ASL
064B: 18       01020  CLC
064C: 65 E0    01030  ADC GCHAR
064E: 85 E6    01040  STA DSTART
0650: A5 E1    01050  LDA GCHAR+1
0652: 69 02    01060  ADC #2
0654: 85 E7    01070  STA DSTART+1
               01080 ;
0656: A0 07    01090  LDY #7
               01100 EXCHAN
0658: B1 E2    01110  LDA (SSTART),Y
065A: 91 E6    01120  STA (DSTART),Y
065C: 88       01130  DEY
065D: 10 F9    01140  BPL EXCHAN
               01150 ;
065F: E6 E8    01160  INC ANZAHL
0661: A5 E8    01170  LDA ANZAHL
0663: C9 07    01180  CMP #7
0665: F0 10    01190  BEQ EXIT
               01200 ;
0667: A5 E2    01210  LDA SSTART
0669: 18       01220  CLC
066A: 69 08    01230  ADC #8
066C: 85 E2    01240  STA SSTART
066E: A5 E3    01250  LDA SSTART+1
0670: 69 00    01260  ADC #0
0672: 85 E3    01270  STA SSTART+1
0674: 4C 43 06 01280  JMP EXCHAN1
               01290 ;
               01300 ; VERLASSEN DES PROGRAMMES
               01310 ;
               01320 EXIT
0677: 60       01330  RTS
               01340 ;
               01350 ; BITFELD FUER DIE ZU
               01360 ; ERSETZENDE ZEICHEN
               01370 ;
               01380 CHARS
0678: 66 18 3C
067B: 66 66 7E
067E: 66 00    01390  .HS 66183C66667E6600
0680: 00 66 00
0683: 66 66 66
0686: 3E 00    01400  .HS 0066006666663E00
0688: 66 3C 66
068B: 66 66 66
068E: 3C 00    01410  .HS 663C666666663C00
0690: 66 00 3C
0693: 66 66 66
0696: 3C 00    01420  .HS 66003C6666663C00
0698: 66 00 3C
069B: 06 3E 66
069E: 3E 00    01430  .HS 66003C063E663E00
06A0: 66 00 66
06A3: 66 66 66
06A6: 7E 00    01440  .HS 6600666666667E00
06A8: 00 3C 66
06AB: 7C 66 7C
06AE: 60 00    01450  .HS 003C667C667C6000
06B0: 3C 66 7C
06B3: 66 7C 60
06B6: 00 00    01460  .HS 3C667C667C600000
               01470 ;
               01480 ; TABELLE FUER POSITION
               01490 ; DER ZEICHEN
               01500 ;
               01510 CHRPOS
06B8: 01 09 0F
06BB: 10 13 15
06BE: 1A       01520  .HS 01090F1013151A

    --- Symbol table ---
00E8: ANZAHL
0678: CHARS
06B8: CHRPOS
00E6: DSTART
0658: EXCHAN
0643: EXCHAN1
0677: EXIT
00E0: GCHAR
0623: LOOP
0637: LOOP1
006A: RAMTOP
00E4: SEND
00E2: SSTART
0601: START

Ok.
```

Listing. Für Assemblerfreaks: Umlaute und »ß«

```
100 REM DEUTSCHER ZEICHENSATZ                <HS>
110 REM ====================                 <AN>
120 REM                                      <AM>
130 I=1536                                   <II>
140 READ J:IF J=256 THEN 160                 <PN>
150 POKE I,J:I=I+1:GOTO 140                  <QD>
160 POKE 106,PEEK(106)-4:GRAPHICS 0          <JE>
170 A=USR(1536)                              <YU>
180 POKE 756,PEEK(106)                       <AK>
190 END                                      <YE>
200 DATA 104,165,106,133,225,169,0           <XT>
210 DATA 133,224,169,0,133,226,169           <HM>
220 DATA 224,133,227,169,0,133,228           <CK>
230 DATA 169,228,133,229,165,224,133         <MG>
240 DATA 230,165,225,133,231,160,0           <KX>
250 DATA 177,226,145,230,200,208,249         <IK>
260 DATA 165,227,197,229,240,7,230           <FM>
270 DATA 227,230,231,76,35,6,169             <MX>
280 DATA 0,133,232,169,120,133,226           <VL>
290 DATA 169,6,133,227,164,232,185           <KS>
300 DATA 184,6,10,10,10,24,101               <QI>
310 DATA 224,133,230,165,225,105,2           <PL>
320 DATA 133,231,160,7,177,226,145           <FK>
330 DATA 230,136,16,249,230,232,165          <QK>
340 DATA 232,201,7,240,16,165,226            <LQ>
350 DATA 24,105,8,133,226,165,227            <SO>
360 DATA 105,0,133,227,76,67,6               <TH>
370 DATA 96,102,24,60,102,102,126            <ZN>
380 DATA 102,0,0,102,0,102,102               <HX>
390 DATA 102,62,0,102,60,102,102             <DM>
400 DATA 102,102,60,0,102,0,60               <MR>
410 DATA 102,102,102,60,0,102,0              <OD>
420 DATA 60,6,62,102,62,0,102                <QX>
430 DATA 0,102,102,102,102,126,0             <BQ>
440 DATA 0,60,102,124,102,124,96             <WB>
450 DATA 0,60,102,124,102,124,96             <WD>
460 DATA 0,0,1,9,15,16,19                    <BZ>
470 DATA 21,26,0,256                         <VK>
```

Listing. Umlaute und »ß« in eigenen Programmen

Programm im Hintergrund

In der Interrupt-Verarbeitung besitzt kaum ein Computer so viele Fähigkeiten wie der Atari XL. So kann man zum Beispiel im Takt des Bildschirmaufbaus ein Interrupt auslösen und damit ein Programm steuern.

Jeder Assembler-Programmierer wird es bestätigen: Interrupts bei Mikroprozessoren sind eine feine Sache. Bestes Beispiel für einen Interrupt ist eine Tastaturabfrage. Man läßt den Prozessor an irgendeiner Aufgabe still vor sich hin arbeiten. Erst wenn eine Taste gedrückt ist, wird der Prozessor von seiner Aufgabe losgerissen, um die Taste abzufragen und ihren Code zum Beispiel in einem Tastaturpuffer zu speichern. Ohne Interrupt könnte der Prozessor nichts anderes tun, als ständig in einer Programmschleife die einzelnen Tasten zu überprüfen. Der Prozessor könnte also nur diese eine Aufgabe erledigen, und solange keine Taste gedrückt wird, würde er keine effektive Arbeit verrichten. Die Rechenzeit wäre verschenkt.

Damit ein Programm aber auch auf relativ seltene Ereignisse reagiert – wie zum Beispiel den Druck auf eine Taste – ohne dabei einzig und allein diese Aufgabe wahrzunehmen, haben die Entwickler der Mikroprozessoren die Interrupts eingebaut. Der 6502 in den Atari-Computern hat davon gleich vier verschiedene. Zum ersten gibt es den Reset. Mit ihm wird dem Prozessor mitgeteilt, wo er beim Einschalten des Computers sein Programm findet. Zum zweiten hat der 6502 einen Interrupt eingebaut, der sich IRQ nennt, **I**nterrupt **Req**uest, auf Deutsch: Jetzt soll ein Interrupt ausgeführt werden. Im Gegensatz zum Reset läßt sich der IRQ abschalten.

Der dritte Interrupt ist als Hilfsmittel für Programmierer gedacht. Beim 6502 gibt es einen Assembler-Befehl, der ebenfalls einen Interrupt auslöst: »BRK«, **Break**. Damit läßt sich bei der Fehlersuche das Programm an kritischen Stellen unterbrechen, um so Hinweise auf einen Fehler zu bekommen.

Der letzte Interrupt beim 6502 ist bei den Atari-Computern gleichzeitig der interessanteste; gemeint ist der **N**on **M**askable **I**nterrupt, der »NMI«. Wie der Name schon sagt, läßt er sich vom Prozessor nicht abschalten, wenn also ein NMI auftritt, muß der Prozessor darauf reagieren.

Bei Atari hat dieser Interrupt zwei Funktionen: Zum einen wird damit ein sogenannter Display List Interrupt ausgeführt (wird an anderer Stelle in dieser Ausgabe besprochen). Zum anderen gibt es im Atari den »Vertical Blank Interrupt«: VBLANK-Interrupt. Jedesmal, wenn im Fernsehgerät oder Monitor der Elektronenstrahl anfängt, ein neues Bild zu zeichnen, wird dieser Interrupt ausgelöst. Im Betriebssystem des Computers liegt nun eine spezielle Interrupt-Routine, die dadurch aufgerufen wird und verschiedene Aufgaben übernimmt: die VBLANK-Routine.

Dazu ein Beispiel: Schalten Sie Ihren Computer ein. Sie befinden sich jetzt im eingebauten Basic. Geben Sie nun folgenden Befehl ein:

```
SETCOLOR 2,0,0
```

Nachdem Sie <RETURN> gedrückt haben, wird der Bildschirm schwarz und die Schrift zeichnet sich hellgrau ab. Bis hierhin verläuft alles wie gewohnt. Der Basic-Interpreter erkennt in »Setcolor« den Befehl, die Farbe auf dem Bildschirm zu verändern. Dazu schaut er sich die Zahlen hinter dem Befehl an, wandelt sie um und schreibt sie in eine spezielle Speicherstelle, die für die Farbe verantwortlich ist. Nun könnte man auf die Idee kommen, selbst direkt diese Speicherstelle mit Hilfe des »POKE«-Befehls zu verändern. Schließlich braucht der Interpreter dann nicht erst die Werte hinter dem Setcolor-Befehl umzurechnen, und man spart dadurch Zeit. Machen wir das einmal:

```
POKE 53272,50
```

Die Speicherstelle 53272 ist dabei für die Hintergrundfarbe beim Textmodus verantwortlich, und der Wert 50 entspricht der Farbe rot. Theoretisch müßte der Bildschirm nun von schwarz auf rot umschalten.

Mitnichten: Der Bildschirm bleibt schwarz. Nachdem Sie <RETURN> gedrückt haben, flackert der Bildschirm bestenfalls kurz rötlich auf. Wie erklärt sich das?

Hier kommt die VBLANK-Routine ins Spiel. Diese holt sich aus einer bestimmten Speicherstelle im Computer einen Wert und schreibt diesen dann in die Speicherstelle, die für die Farbe verantwortlich ist. Da die VBLANK-Routine im Takt des Bildschirmaufbaus aufgerufen wird, alle 50stel Sekunde, wird die Speicherstelle mit der Bildschirmfarbe in kurzen Zeitabständen neu beschrieben. Die Speicherstelle, aus der der Wert geholt wird, nennt sich »Schattenregister«. Wenn man also die Bildschirmfarbe direkt umschalten will, muß man den Farbwert direkt in das entsprechende Schattenregister schreiben. Sie werden sich jetzt mit Recht fragen, wozu dieser ganze Aufwand gut sein soll.

Um bei diesem Beispiel zu bleiben: Würde es den VBLANK-Interrupt nicht geben, und würde man die Farbe direkt umschalten, so würde dies meist dann geschehen, wenn sich der Elektronenstrahl gerade mitten auf dem Bildschirm befindet. Das Bild würde kurz flackern. Um dies zu vermeiden, schreibt man den Wert in ein Schattenregister, und erst beim nächsten VBLANK wird die Bildschirmfarbe umgeschaltet.

Mit folgendem Befehl verändert man das Schattenregister:

```
POKE 710,50
```

Sobald Sie <RETURN> drücken, wird der Bildschirm rot. Speicherstelle 710 ist also das Schattenregister für die Hintergrundfarbe beim Textmodus. Für die anderen Setcolor-Befehle gibt es entsprechende Register: Setcolor 0 hat Speicherstelle 708, Setcolor 1 hat Speicherstelle 709, Setcolor 3 Speicherstelle 711 und Setcolor 4 hat Speicherstelle 712.

Mit den Speicherstellen, die direkt die Farbe umschalten, läßt sich noch eine nette Spielerei durchführen: Geben Sie folgendes kleine Programm genau so ein, wie es hier gedruckt ist, und lassen Sie es laufen.

```
100 POKE 53272,0
110 REM
120 REM
130 POKE 53272,50
140 GOTO 100
```

Ist das für reines Basic nicht ein netter Effekt? Spaßeshalber können Sie Zeile 110 löschen und das Programm dann laufen lassen.

Die VBLANK-Routine hat noch weitere Funktionen. Wußten Sie zum Beispiel, daß der Atari in den Speicherstellen 18, 19 und 20 eine Uhr eingebaut hat? Speicherstelle 18 wird im 50stel Sekundentakt erhöht. Wenn diese dann wieder bei Null anfängt, wird Speicherstelle 19 erhöht, und deren Überlauf in Speicherstelle 20 registriert. So lassen sich mit einfachen Mitteln in eigene Programme Zeitabfragen einbauen.

Über die Uhr wird noch eine weitere interessante Funktion des Computers gesteuert. Sie tritt jedesmal dann in Aktion, wenn beim Computer länger als zirka sieben Minuten keine Taste mehr betätigt wurde. Dann nämlich beginnt der Computer die Farben zum Schutz des Bildschirms völlig zufällig zu verändern. Dieser »Attrakt Mode« genannte Zustand beginnt immer dann, wenn Speicherstelle 77, die im Takt von Speicherstelle 20 erhöht wird, den Wert 127 überschreitet. Bei jedem Tastendruck wird Speicherstelle 77 wieder auf Null gesetzt, und die Farben hören auf, sich zu verändern. Geben Sie folgende Zeile ein:

```
POKE 77,128
```

Nachdem Sie <RETURN> gedrückt haben, schalten die Farben um.

Neben diesen Funktionen führt die VBLANK-Routine noch viele weitere Aufgaben aus. Zum Beispiel wird der Joystickanschluß ausgelesen und in die Speicherstelle 278 für Joystick 0 und Speicherstelle 279 für Joystick 1 übertragen.

Die VBLANK-Routine übernimmt allgemein alle Aufgaben, bei denen es egal ist, ob sie sofort oder höchstens eine 50stel Sekunde später ausgeführt werden. Da der VBLANK-Interrupt immer im nicht-sichtbaren Teil des Bildschirms ausgeführt wird, wäre es schön, wenn man den Interrupt für eigene Aufgaben einsetzen könnte. Zum Beispiel ließe sich bei der Player-Missile-Grafik damit ein Player ruckfrei über den Bildschirm bewegen.

Voraussetzung dafür ist aber, daß man sich in der Maschinensprache des 6502 gut auskennt, denn die VBLANK-Interrupts sind extrem zeitkritisch und müssen unbedingt in Maschinensprache programmiert werden. Ganz nebenbei ist es auch sehr kompliziert, zum Beispiel ein Basic-Programm von einem VBLANK-Interrupt aus aufzurufen.

Zunächst einmal muß festgelegt werden, wo die VBLANK-Interrupt-Routine stehen soll. Dann teilt man dem Computer mit, wo sich diese Rou-

```
Ok.

             00010 ;**********************
             00020 ;*                    *
             00030 ;* BLASTSOUND         *
             00040 ;*                    *
             00050 ;* 12.1.1986          *
             00060 ;*                    *
             00070 ;* Henrik Fisch       *
             00080 ;*                    *
             00090 ;**********************
             00100 ;
             00110 ; Dieses Programm simuliert
             00120 ; den Sound eines bekannten
             00130 ; und beruehmten Spiels
             00140 ;
             00150 ; OS - EQUATES
             00160 ;
0224:        00170 VVBLKD .EQ $224
E462:        00180 XITVBV .EQ $E462
E474:        00190 WARMSV .EQ $E474
             00200 ;
             00210 ; HARWARE REGISTER
             00220 ;
D200:        00230 AUDF1  .EQ $D200
D201:        00240 AUDC1  .EQ $D201
D20A:        00250 RANDOM .EQ $D20A
D40E:        00260 NMIEN  .EQ $D40E
             00270 ;
             00280 ; VON DIESEM PROGRAMM BENUTZT
             00290 ;
0080:        00300 SNDADR .EQ $80
0082:        00310 VOLADR .EQ $82
             00320 ;
             00330 ; PROGRAMMSPEICHERUNG
             00340 ;
             00350  .OR $2000
             00360  .TA $2000
             00370  .TF "D:BLASTSND.COM"
             00380 ;
             00390 ; INITIALISIERUNG
             00400 ;
             00410 START
2000: A9 00   00420  LDA #0
2002: 8D 0E D4 00430  STA NMIEN
2005: A9 1A   00440  LDA #VBLANK
2007: 8D 24 02 00450  STA VVBLKD
200A: A9 20   00460  LDA /VBLANK
200C: 8D 25 02 00470  STA VVBLKD+1
200F: A9 40   00480  LDA #$40
2011: 8D 0E D4 00490  STA NMIEN
2014: A9 06   00500  LDA #6
2016: 8D 7B 20 00510  STA VOLPNT
2019: 60      00520  RTS
             00530 ;
             00540 ; HAUPTPROGRAMM
             00550 ;
             00560 VBLANK
201A: AD 7B 20 00570  LDA VOLPNT
201D: C9 06   00580  CMP #6
201F: D0 49   00590  BNE VBLANK1
             00600 ;
             00610 ; AUSWAEHLEN DES SOUND'S
             00620 ;
2021: AD 0A D2 00630  LDA RANDOM
2024: 29 03   00640  AND #3
2026: 85 80   00650  STA SNDADR
2028: AD 0A D2 00660  LDA RANDOM
202B: 29 02   00670  AND #2
202D: 18      00680  CLC
202E: 65 80   00690  ADC SNDADR
2030: 0A      00700  ASL
2031: 18      00710  CLC
2032: 69 7D   00720  ADC #SOUNDS
2034: 85 80   00730  STA SNDADR
2036: A9 00   00740  LDA #0
2038: 69 20   00750  ADC /SOUNDS
203A: 85 81   00760  STA SNDADR+1
203C: A0 00   00770  LDY #0
203E: B1 80   00780  LDA (SNDADR),Y
2040: 8D 7C 20 00790  STA DISTOR
2043: C8      00800  INY
2044: B1 80   00810  LDA (SNDADR),Y
2046: 8D 00 D2 00820  STA AUDF1
             00830 ;
             00840 ; AUSWAEHLEN DER
             00850 ; LAUTSTAERKETABELLE
             00860 ;
2049: AD 0A D2 00870  LDA RANDOM
204C: 29 01   00880  AND #1
204E: 85 82   00890  STA VOLADR
2050: 0A      00900  ASL
2051: 85 83   00910  STA VOLADR+1
2053: A5 82   00920  LDA VOLADR
2055: 0A      00930  ASL
2056: 0A      00940  ASL
2057: 18      00950  CLC
2058: 65 83   00960  ADC VOLADR+1
205A: 18      00970  CLC
205B: 69 89   00980  ADC #VOLTAB
205D: 85 82   00990  STA VOLADR
205F: A9 00   01000  LDA #0
2061: 69 20   01010  ADC /VOLTAB
2063: 85 83   01020  STA VOLADR+1
2065: A9 00   01030  LDA #0
2067: 8D 7B 20 01040  STA VOLPNT
             01050 ;
             01060 ; TON ERZEUGEN
             01070 ;
             01080 VBLANK1
206A: AC 7B 20 01090  LDY VOLPNT
206D: B1 82   01100  LDA (VOLADR),Y
206F: 0D 7C 20 01110  ORA DISTOR
2072: 8D 01 D2 01120  STA AUDC1
             01130 ;
             01140 ; LAUTSTAERKE VERAENDERN
             01150 ;
2075: EE 7B 20 01160  INC VOLPNT
2078: 4C 62 E4 01170  JMP XITVBV
             01180 ;
             01190 ; HILFSSPEICHERSTELLE
             01200 ;
207B: 00      01210 VOLPNT  .HS 00
207C: 00      01220 DISTOR  .HS 00
             01230 ;
             01240 ; SOUNDTABELLE
```

Assemblerlisting »Blastsound« für alle, die's genau wissen wollen

```
                 01250 ;
                 01260 SOUNDS
207D: 20 05      01270  .HS 2005
207F: 20 07      01280  .HS 2007
2081: 80 02      01290  .HS 8002
2083: 80 03      01300  .HS 8003
2085: 80 05      01310  .HS 8005
2087: 80 FF      01320  .HS 80FF
                 01330 ;
                 01340 ; VOLUMENTABELLE
                 01350 ;
                 01360 VOLTAB
2089: 0E 0C 0A
208C: 08 06 04   01370  .HS 0E0C0A080604
208F: 08 06 04
2092: 02 00 00   01380  .HS 080604020000
                 01390 ;
                 01400 ; STARTADRESSE
                 01410 ; DES PROGRAMMS
                 01420 ;
                 01430  .OR $2E2
02E2: 00 20      01440  .DA START

      --- Symbol table ---
D201: AUDC1
D200: AUDF1
207C: DISTOR
D40E: NMIEN
D20A: RANDOM
0080: SNDADR
207D: SOUNDS
2000: START
201A: VBLANK
206A: VBLANK1
0082: VOLADR
207B: VOLPNT
2089: VOLTAB
0224: VVBLKD
E474: WARMSV
E462: XITVBV

Ok.
```

Assemblerlisting »Blastsound« (Schluß)

```
Programmname :D:BLASTSND.COM
Laenge       :160 Bytes

0000:FF FF 00 20 95 20 A9 00 <30>
0008:8D 0E D4 A9 1A 8D 24 02 <A0>
0010:A9 20 8D 25 02 A9 40 8D <A3>
0018:0E D4 A9 06 8D 7B 20 60 <82>
0020:AD 7B 20 C9 06 D0 49 AD <D3>
0028:0A D2 29 03 85 80 AD 0A <7C>
0030:D2 29 02 18 65 80 0A 18 <76>
0038:69 7D 85 80 A9 00 69 20 <EA>
0040:85 81 A0 00 B1 80 8D 7C <15>
0048:20 C8 B1 80 8D 00 D2 AD <25>
0050:0A D2 29 01 85 82 0A 85 <C2>
0058:83 A5 82 0A 0A 18 65 83 <B2>
0060:18 69 89 85 82 A9 00 69 <23>
0068:20 85 83 A9 00 8D 7B 20 <C1>
0070:AC 7B 20 B1 82 0D 7C 20 <8D>
0078:8D 01 D2 EE 7B 20 4C 62 <6B>
0080:E4 00 00 20 05 20 07 80 <2B>
0088:02 80 03 80 05 80 FF 0E <34>
0090:0C 0A 08 06 04 08 06 04 <CA>
0098:02 00 00 00 00 00 00 00 <E9>
```

»Blastsound« als AMPEL-Listing

```
Ok.

                 00010 ;****************
                 00020 ;*              *
                 00030 ;* FARBDIMMER   *
                 00040 ;*              *
                 00050 ;* 22.7.1987    *
                 00060 ;*              *
                 00070 ;* Henrik Fisch *
                 00080 ;*              *
                 00090 ;****************
                 00100 ;
                 00110 ; Dieses Programm wechselt
                 00120 ; im VBLANK-Interupt
                 00130 ; die Hitergrundfarbe
                 00140 ; eines GRAPHICS 0-Bildschirms
                 00150 ;
                 00160 ; HARDWARE REGISTER
                 00170 ;
D018:            00180 COLPF2 .EQ $D018
D40E:            00190 NMIEN  .EQ $D40E
                 00200 ;
                 00210 ; OS - EQUATES
                 00220 ;
0224:            00230 VVBLKD .EQ $224
E462:            00240 XITVBV .EQ $E462
                 00250 ;
                 00260 ; PROGRAMMABSPEICHERUNG
                 00270 ;
                 00280  .OR $600
                 00290  .TA $600
                 00300  .TF "D:FARBDIM.COM"
                 00310 ;
                 00320 ; INITIALISIERT DIE
                 00330 ; VBLANK-ROUTINE
                 00340 ;
                 00350 INIT
0600: A9 00      00360  LDA #0
0602: 8D 0E D4   00370  STA NMIEN
0605: A9 15      00380  LDA #VBLANK
0607: 8D 24 02   00390  STA VVBLKD
060A: A9 06      00400  LDA /VBLANK
060C: 8D 25 02   00410  STA VVBLKD+1
060F: A9 40      00420  LDA #$40
0611: 8D 0E D4   00430  STA NMIEN
0614: 60         00440  RTS
                 00450 ;
                 00460 ; VBLANK-ROUTINE
                 00470 ; WIRD BEI JEDEM VBI
                 00480 ; ANGESPRUNGEN
                 00490 ;
                 00500 VBLANK
0615: AD 5F 06   00510  LDA RICHTG
0618: 30 1D      00520  BMI DUNKLR
                 00530 ;
                 00540 ; FARBE WIRD EINE
                 00550 ; STUFE HELLER GESTELLT
                 00560 ;
                 00570 HELLER
061A: AD 5E 06   00580  LDA FARBE
061D: 29 0E      00590  AND #$E
061F: C9 0E      00600  CMP #$E
0621: D0 08      00610  BNE HELLER1
0623: A9 80      00620  LDA #$80
0625: 8D 5F 06   00630  STA RICHTG
0628: 4C 52 06   00640  JMP DUNKLR1
                 00650 HELLER1
062B: EE 5E 06   00660  INC FARBE
062E: AD 5E 06   00670  LDA FARBE
0631: 8D 18 D0   00680  STA COLPF2
0634: 4C 5B 06   00690  JMP EXIT
                 00700 ;
                 00710 ; FARBE WIRD EINE
                 00720 ; STUFE DUNKLER GESTELLT
                 00730 ;
                 00740 DUNKLR
0637: AD 5E 06   00750  LDA FARBE
063A: 29 0E      00760  AND #$E
063C: D0 14      00770  BNE DUNKLR1
063E: A9 00      00780  LDA #0
0640: 8D 5F 06   00790  STA RICHTG
0643: AD 5E 06   00800  LDA FARBE
0646: 18         00810  CLC
0647: 69 10      00820  ADC #$10
0649: 8D 5E 06   00830  STA FARBE
064C: 8D 18 D0   00840  STA COLPF2
064F: 4C 5B 06   00850  JMP EXIT
                 00860 DUNKLR1
0652: CE 5E 06   00870  DEC FARBE
0655: AD 5E 06   00880  LDA FARBE
0658: 8D 18 D0   00890  STA COLPF2
                 00900 ;
                 00910 ; VERLASSE VBLANK-ROUTINE
                 00920 ;
                 00930 EXIT
065B: 4C 62 E4   00940  JMP XITVBV
                 00950 ;
                 00960 ; VON DIESEM PROGRAMM BENUTZT
                 00970 ;
065E: 00         00980 FARBE  .HS 00
065F: 00         00990 RICHTG .HS 00
                 01000 ;
                 01010 ; STARTADRESSE
                 01020 ;
                 01030  .OR $2E2
02E2: 00 06      01040  .DA INIT

      --- Symbol table ---
D018: COLPF2
0637: DUNKLR
0652: DUNKLR1
065B: EXIT
065E: FARBE
061A: HELLER
062B: HELLER1
0600: INIT
D40E: NMIEN
065F: RICHTG
0615: VBLANK
0224: VVBLKD
E462: XITVBV

Ok.
```

Assemblerlisting zum »Farbdimmer«

```
Programmname :D:FARBDIM.COM
Laenge       :104 Bytes

0000:FF FF 00 06 60 06 A9 00 <7C>
0008:8D 0E D4 A9 15 8D 24 02 <7C>
0010:A9 06 8D 25 02 A9 40 8D <2A>
0018:0E D4 60 AD 5F 06 30 1D <69>
0020:AD 5E 06 29 0E C9 0E D0 <11>
0028:08 A9 80 8D 5F 06 4C 52 <14>
0030:06 EE 5E 06 AD 5E 06 8D <52>
0038:18 D0 4C 5B 06 AD 5E 06 <3D>
0040:29 0E D0 14 A9 00 8D 5F <52>
0048:06 AD 5E 06 18 69 10 8D <D2>
0050:5E 06 8D 18 D0 4C 5B 06 <5E>
0058:CE 5E 06 AD 5E 06 8D 18 <5E>
0060:D0 4C 62 E4 00 00 00 00 <DD>
```

»Farbdimmer« im AMPEL-Format

tine befindet. Das macht man über die Speicherstellen 546 und 547 ($222 und $223) oder 548 und 549 ($224 und $225). Der Computer springt bei einem VBLANK-Interrupt nacheinander zu den beiden Adressen, die er in diesen Speicherstellen liest. In den ersten beiden Speicherstellen steht die Adresse der eingebauten VBLANK-Routine. Die zweite Adresse ist normalerweise Nebensache; mit ihr läßt sich ganz einfach eine eigene Routine direkt an die eingebaute VBLANK-Routine anhängen.

In 548 und 549 wird also die Startadresse der eigenen Routine geschrieben. Vorher sollte man jedoch den Interrupt abschalten, denn wenn der Interrupt ausgerechnet dann auftritt, wenn das Low-Byte der Adresse bereits in 548 steht, das High-Byte jedoch noch nicht übertragen ist, springt der Computer zu dieser falschen Speicherstelle und hängt sich meistens auf.

Wahrscheinlich werden Sie sich jetzt fragen, wie Sie einen »Non Maskable Interrupt« abschalten sollen. Der Atari hat jedoch ein Register, mit dem sich auch dieser Interrupt verhindern läßt. Die Speicherstelle liegt bei Adresse 54286 ($D40E). Löscht man hier Bit 6, so ist der VBLANK-Interrupt abgeschaltet. Erst jetzt setzt man die Adresse auf die eigene Routine.

Nachdem die VBLANK-Routine abgeschlossen ist, soll der Programmablauf wieder aufgenommen werden. Dafür gibt es einen speziellen Maschinenbefehl des 6502, den »RTI«, »Return from Interrupt«. Dieser darf jedoch nicht verwendet werden, denn bei jedem VBLANK-Interrupt sichert das Betriebssystem die Register des 6502 auf dem Stack. Diese stehen damit dem Programmierer für die Dauer der Routine allesamt zur Verfügung. Nachdem die Routine ihre Arbeit beendet hat, müssen diese Registerinhalte jedoch wieder restauriert werden, damit das unterbrochene Programm ohne Probleme weiterarbeitet. Für diese Aufgabe gibt es im Betriebssystem ebenfalls eine Routine. Sie nennt sich »XITVBV« und wird aufgerufen, indem man in Assembler zur Adresse 58466 ($E462) springt. Dieses Programm verläßt danach die Interrupt-Routine.

Die beiden Programme demonstrieren den VBLANK-Interrupt. Das erste Programm läßt im Textmodus nacheinander alle Farben aufleuchten. Das zweite Programm läßt eine Klangfolge ertönen, die einem bekannten Spiel nachempfunden ist. Wie beim VBLANK-Interrupt üblich, lassen sich trotzdem alle Funktionen des Computers weiterhin benutzen. Anhand der Assembler-Listings sehen Sie, wie die Programme funktionieren. Die Programme lassen sich nur mit der <RESET>-Taste abschalten. (hf)

Wichtige Systemvariablen

Name	Adresse Dezimal	Adresse Hexadezimal	Erklärung
ATRACT	707	4D	Werte größer 127 leiten den Attrakt-Mode ein
COLOR0	708	2C4	Entspricht SETCOLOR 0,...
COLOR1	709	2C5	Entspricht SETCOLOR 1,...
COLOR2	710	2C6	Entspricht SETCOLOR 2,...
COLOR3	711	2C7	Entspricht SETCOLOR 3,...
COLOR4	712	2C8	Entspricht SETCOLOR 4,...
COLBK	53274	D01A	Farbregister für COLOR4
COLPF0	53270	D016	Farbregister für COLOR0
COLPF1	53271	D017	Farbregister für COLOR1
COLPF2	53272	D018	Farbregister für COLOR2
COLPF3	53273	D019	Farbregister für COLOR3
NMIEN	54286	D40E	Wenn Bit 6 gelöscht ist, wird der VBLANK-Interrupt abgeschaltet
RTCLOK	18	12	System-Timer, wird in 50stel-Sekunden-Schritten erhöht
	19	13	Überlauf von RTCLOK Speicherstelle 18
	20	14	Überlauf von RTCLOK Speicherstelle 19
VVBLKD	548,549	224,225	Beginn einer VBLANK-Routine, die nach der Systemroutine ausgeführt wird
VVBLKI	546,547	222,223	Beginn der System-VBLANK-Routine
XITVBV	58466	E462	Beendet eine VBLANK-Routine

Neues aus den USA

Wenn auch die kleinen Atari-Computer in Deutschland relativ stark vernachlässigt werden, in den USA ist dies nicht so. Die interessantesten neuen Produkte stellen wir hier kurz vor.

An Programmiersprachen für die Atari-Computer herrscht bislang kein Mangel. Von Basic, von dem es bislang sieben verschiedene Interpreter gibt, über Pascal zu Assembler wird alles angeboten, was das Herz begehrt. Sogar Forth und die KI-Sprache Lisp sind für die kleinen Computer erhältlich. Nur eine Sprache steht bislang nicht in einer vernünftigen Implementation zur Verfügung: C. Gut, es gibt den »Deep Blue C«-Compiler von Antic und den »C/65«-Compiler von OSS. Beide Programme weisen aber den Nachteil auf, daß sie weder die Standardfunktionen von C enthalten, noch daß sie besonders Speicherplatz-optimierend arbeiten. Zudem compilieren sie ein Programm gähnend langsam.

Der lichtschnelle C-Compiler

Es grenzt fast an ein Wunder, daß doch noch eine Firma einen C-Compiler entwickelte und ihn für die kleinen Atari-Computer verfügbar machte. Das Produkt nennt sich »Lightspeed C«-Compiler von Clearstar Softechnologies. Das Programm wird auf einer Diskette geliefert und benötigt 48 KByte Speicher. Es läuft nicht nur auf den XL/XE-Computern, sondern auch auf den alten 400er/800er-Computern von Atari.

Soviel vorweg: In dem Compiler ist der gesamte von Kernighan und Ritchie vorgeschlagene Sprachumfang enthalten. Damit hat man eine moderne Sprache zur Verfügung, die vollständig kompatibel zu anderen Computern ist. Auch wenn es sich lächerlich anhört, aber ein in C geschriebenes Programm läuft auf Computern wie Atari ST, Amiga oder IBM-PC genauso wie auf Mini- und Großcomputern wie VAX und Cray, und dank des Lightspeed-Compilers nun auch auf dem kleinen Atari. Eine Veränderung muß man allerdings in Kauf nehmen. Da der Atari die geschweiften Klammern nicht kennt – diese sind bei jeder Funktionsdefinition in C notwendig – werden diese durch Klammern mit vorangestelltem Dollar-Zeichen ersetzt.

Selbstverständlich arbeitet das Programm mit einer in den 130 XE eingebauten RAM-Disk zusammen. Die Compilier- und Linkzeiten verringern sich dadurch weiter. Dazu ein paar Daten: Der Lightspeed-C-Compiler benötigt zum Compilieren eines Primzahlen-Programms, das als Source-Text auf der Diskette 1 KByte belegte, eine Minute. Dabei entsteht ein Maschinenprogramm, das nur 3 Sektoren auf der Diskette beansprucht. Diese Zeiten und Programmlängen lassen sich mit einem mitgelieferten »Optimizer« weiter verbessern.

Tatsache ist, daß mit dem Lightspeed-C-Compiler auf dem Atari eine moderne Programmiersprache zur Verfügung steht, die den Vergleich mit anderen Implementationen auf größeren Computern nicht zu scheuen braucht. Außerdem läßt sich jetzt schon ein wenig im voraus üben, wenn später auf ein größeres System umgestiegen werden soll, wie zum Beispiel auf einen Atari ST oder einen Amiga. Der Compiler ist für 39,95 Dollar bei folgender Adresse erhältlich:
Clearstar Softechnologies
1501 Wood Avenue 36
Sumner, WA 98390
U.S.A.

RAM-Disk, paralleler Port und Modem-Anschluß

Ein weiteres interessantes Produkt stellt die Firma ICD her. Diese Firma sollte man als Atari-8-Bit-Enthusiast sowieso im Auge behalten. Das Produkt nennt sich »MIO-Board« und wird an den parallelen Bus der XL-Computer angeschlossen.

Zum Beispiel ist auf der Platine ein Centronics-Port und ein RS232-kompatibler serieller Anschluß enthalten. Damit lassen sich Modems betreiben (in Deutschland noch nicht zugelassen), oder Drucker anschließen, die diese Art der Datenübertragung verlangen. Am interessantesten ist jedoch die RAM-Disk, die in einer Version von 256 KByte und 1 MByte erhältlich ist. Die RAM-Disk wird von einem mitgelieferten DOS unterstützt und läßt sich beliebig in verschiedene Laufwerke aufteilen. Das Besondere an der RAM-Disk ist, daß sich der Computer ruhig aufhängen darf. Bei einer normalen RAM-Disk ist deren Inhalt dann verloren. Da das MIO-Board aber eine separate Stromversorgung aufweist, ist ihr Inhalt auch nach dem Ein- und Ausschalten des Computers noch vorhanden.

Die letzte Eigenschaft des MIO-Boards ist zugleich auch die erstaunlichste Einrichtung. Es lassen sich bis zu 8 Harddisk-Laufwerke mit jeweils bis zu 16 MByte Speicherkapazität anschließen. Auch wenn diese Speicherkapazität für die kleinen Computer unsinnig erscheint, so manches Mal wünscht man sich doch eine Festplatte, besonders dann, wenn bei einem Programmiervorhaben zum hundertsten Mal die Diskette gewechselt wird.
Das MIO-Board ist bei
ICD
1220 Rock Street
Rockford, IL 61101-1437
U.S.A.
für 199 Dollar mit 256 KByte und für 349 Dollar mit 1 MByte erhältlich.

Universelles Malprogramm

Wenn ein neues Malprogramm vorgestellt wird, muß es schon einiges können. Und wenn es sich durchsetzen soll, muß es wenigstens zu einem der drei bisher meistverkauften Malprogramme kompatibel sein (»Micro Painter» von Datasoft, »Atari Artist« von Atari oder »Fun with Art«). »Blazing Paddles« von Baudville hält diesen Anforderungen stand. Es läßt sich mit einem Joystick, einem Grafik-Tablett, einem Lichtgriffel oder sogar mit Drehreglern betreiben.

Blazing Paddles unterstützt im Gegensatz zu anderen Malprogrammen mehrere Grafikstufen. Neben den üblichen Befehlen zum Zeichnen von Linien, Kreisen und Boxen, zum Festlegen der Farben und zum Speichern und Laden von Diskette, besitzt es Befehle zum Laden von Bildern in den Formaten der oben angegebenen Malprogramme. Blazing Paddles ist bei
Baudville
1001 Medical Park Drive SE
Grand Rapids
MI 49506
U.S.A.
für 34,95 Dollar erhältlich. (hf)

Komponieren ohne Noten

Nach Compilern für Programme gibt es in letzter Zeit Compiler, mit denen sich sehr gut klingende Musikstücke entwickeln lassen, die auch in eigene Programme einbindbar sind.

Masic heißt das neueste Programm der Firma Rätz-Eberle. Dabei handelt es sich um einen softwaremäßigen Musiksynthesizer für die kleinen Atari XL/XE-Computer. Solch ein Programm ist vor einiger Zeit schon einmal erschienen. Gemeint ist »The Soundmachine«, mit der sich fantastisch klingende Musikstücke entwicklen und in eigene Programme einbinden lassen. Das Besondere an der Soundmachine ist aber ihre Benutzerfreundlichkeit. So wird fast alles mit dem Joystick gesteuert. Die einzelnen Noten eines Musikstücks lassen sich mit dem Joystick auf richtigen Notenlinien plazieren. Das setzt allerdings voraus, daß man Noten lesen kann. Der Ungeübte weiß mit dieser Darstellungsmethode wenig anzufangen. Masic soll diese Nachteile umgehen, indem für das Programm eine eigene Musik-Programmiersprache entwickelt wurde.

Wenn man Masic bestellt, bekommt man eine Programmdiskette, die das Masic-Programm selbst sowie zwei Hilfsprogramme und eine 28 Seiten starke Anleitung enthält. Auf der Schutzhülle der Diskette ist der Name des Bestsellers zusammen mit einer Codenummer abgedruckt. Diese Nummer muß beim Start des Programms zusammen mit dem Namen in den Computer eingegeben werden. Nur wenn die Nummer und der Name mit den im Programm gespeicherten Daten übereinstimmen, ist Masic funktionsfähig.

Nach diesem etwas kuriosen aber sehr anwenderfreundlichen Kopierschutz – die Programmdiskette ist nicht weiter kopiergeschützt – wird man vom Programm nach der Tastaturbelegung gefragt. Dies erscheint ein bißchen unsinnig. Warum sollte man schließlich eine Tastatur, an die man sich gewöhnt hat, plötzlich umbelegen? Es wird jedoch verständlich, wenn man sich das Konzept von Masic etwas genauer anschaut. Nach dieser Eingabe befindet man sich nämlich im Editor von Masic. Hier wird, ähnlich wie bei einer normalen Programmiersprache, das Musik-Programm eingegeben. Dabei läßt sich das Programm beliebig auf dem Bildschirm gestalten (siehe Bild). In der unteren Zeile des Editors werden ständig die Spalte und Zeile angezeigt, in der sich der Cursor gerade befindet. Schnappt man sich die Anleitung und liest die Bedienung des Programmeditors durch, so wird sofort deutlich, warum der Programmierer von Masic eine deutsche Tastaturbelegung sowie die Spalten- und Zeilenanzeige eingebaut hat: Der Editor von Masic läßt sich gleichzeitig als Texteditor verwenden, zum Beispiel um Briefe zu schreiben. Der Autor des Handbuchs weist auch extra darauf hin, daß das Handbuch mit dem Masic Texteditor geschrieben wurde.

```
; 1. Der Huellkurvenbereich

[
; Huellkurve Nr. 0

 0
 4  8 12 15 14:13 12 11 10  9
 9  8  8  7  7: 7  6  6  6  6
 5  5  5  5  4: 4  4  4  4  3
 3  3  3  3  3: 2  2  2  2  2
 2  2  1  1  1: 1  1  1  1  1
 0  0  0  0  0: 0  0  0  0  0
 0  0  0
]

; 2. Der MASIC-Programmteil

[
marke 1
 t8 ; Alle folgenden Noten
    ; sind Achtelnoten
 e1 a1 e1 b1 e1 g1 a1
 e1 c2 e1 d2 e1 b1 c2
 e1 a1 e1 b1 e1 g1 a1
 e1 c2 e1 d2 e1 b1 c2
 e1 b1
 rts

marke 2
 t8
 c1 f1 c1 g1 c1 e1 f1
 c1 a1 c1 b1 c1 g1 a1
 c1 f1 c1 g1 c1 e1 f1
 c1 a1 c1 b1 c1 g1 a1
 c1 g1
 rts

marke 3
 t8
 g1 c2 g1 d2 g1 b1 c2
 g1 e2 g1 f2 g1 d2 e2
 g1 c2 g1 d2 g1 b1 c2
 g1 e2 g1 f2 g1 d2 e2
 g1 d2
 a1 d2 a1 e2 a1 c2 d2
 a1 f2 a1 g2 a1 e2 f2
 a1 d2 a1 e2 a1 c2 d2
 a1 f2 a1 g2 a1 e2 f2
 a1 e2
 rts

start          ; Beginn des Programms
 relt8 10      ; Grundtempo Achtelnote
 stat 0 0 1    ; Tonkanal 0 einschalten
 klg 0 160     ; Klangart Kanal 0 setzen
 hku 0 0       ; Huellkurve Nr. 0
               ; fuer Kanal 0
 rel 0 0 1     ; Relativmodus der Note
               ; fuer Kanal 0 einschalten
marke 4
 jsr 1         ; Unterprogramm 1 aufrufen
 jsr 1
 jsr 2
 jsr 1
 jsr 3
 jsr 3
 jmp 4         ; Endlosschleife
ende
]
```

Ein bekannte Melodie in Masic

Doch der Hauptzweck des Editors liegt darin, Musikstücke einzugeben und Klänge zu erschaffen. Jedes Masic-Programm besteht aus einem Hüllkurventeil und einem Notenteil. Jeder der beiden Teile wird mit rechteckigen Klammern eingeschlossen. Dazwischen dürfen beliebige Kommentare stehen. In ein Programm lassen sich Kommentare einfügen, indem davor einfach ein Semikolon gestellt wird.

Eine Hüllkurve hat die Aufgabe, den Lautstärkeverlauf eines Klangs zu bestimmen. Ein Klavierklang zum Beispiel wird sofort sehr laut, sinkt dann in der Lautstärke langsam ab und geht fast augenblicklich auf Null zurück, wenn die Taste wieder losgelassen wird. Im Gegensatz dazu ertönt ein Trompetenklang zwar auch sofort in voller Lautstärke, bleibt dann aber für die Dauer des Tons in der Lautstärke beständig. Zum Simulieren dieser Lautstärkeverläufe sind die Hüllkurven da. Da der Soundchip der Atari-Computer maximal 16 Lautstärkewerte verarbeitet, erfolgt die Angabe der Hüllkurven beim Masic auch in diesen 16 Werten. Die Hüllkurve besteht damit im Programm aus einer Folge von Zahlen. Jede Zahl stellt einen Lautstärkewert zu einem bestimmten Zeitpunkt der Hüllkurve dar. Eine Hüllkurve setzt sich dabei aus maximal 255 Werten zusammen. Da das Masic-Programm zum Bestimmen der Notenlängen den VBLANK-Interrupt der Atari-Computer ausnutzt, ist ein Hüllkurvenschritt genau eine 50stel Sekunde lang. Der Vorteil dieser Hüllkurven-Programmierung liegt nun darin, daß nicht nur an- und absteigende Lautstärkeverläufe programmiert werden können. Auch Effekte, wie zum Beispiel das An- und Abschwellen eines Tons bei einem Vibraphon oder das langsame Ansteigen und dann plötzliche Abfallen der Lautstärke bei einem Tonband, das rückwärts läuft, werden erzeugt. Der Nachteil dieser Programmierung besteht darin, daß alle Notenwerte umständlich in den Computer eingetippt werden müssen und daß man bei jeder Änderung erst das ganze Programm neu starten muß, ehe man den Klang hört. Es gehört also etwas Übung dazu, einen Klang richtig schön zu »komponieren«.

Der zweite Programmteil umfaßt die Notenwerte und verschiedene Steuerkommandos, mit denen das Musikstück abgespielt wird. Am Anfang

einer Notenfolge muß eine Sprungmarke stehen, damit das spätere Programm weiß, an welcher Stelle die Noten beginnen. Für jeden Notenwert wird die Dauer eines Tones und dessen Notenhöhe eingegeben. Die Dauer umfaßt dabei eine ganze bis eine 32stel Note, mit allen ihren Zwischenschritten, wie zum Beispiel eine punktierte Achtelnote. Die effektive Dauer einer Note wird später im Programm angegeben. Falls nun eine Folge von Noten die gleiche Länge hat, brauchen die Notenlängen nur am Anfang dieser Folge bestimmt zu werden. Für alle anderen Noten nimmt das Programm dann automatisch diese Notenlänge an. Ein Stück, das nur aus Viertelnoten besteht, benötigt also nur einen Notenlängenbefehl.

Die Noten werden mit ihrem Namen eingegeben. Die Note »H« wird allerdings nach amerikanischem Vorbild mit »B« notiert. Der Tonumfang erstreckt sich dabei über sieben Oktaven.

Auch als Texteditor geeignet

Die Anzahl der zu spielenden Noten eines Stücks ist nur durch den zur Verfügung stehenden Speicherplatz beschränkt. Der Editor stellt dabei maximal 17659 Speicherplätze zur Verfügung. Da der Editor auch zur Textverarbeitung eingesetzt werden kann, beansprucht jedes eingetippte Zeichen ein Byte. In der Anleitung steht über dem zur Verfügung stehenden Speicherplatz eine Bemerkung, die die Freude an Masic ein wenig trübt. Sollte man einmal mehr Zeichen eingegeben haben, als Platz zur Verfügung steht, kann es sein, daß sich das Programm aufhängt. Das ist weniger schön, liegt aber womöglich daran, daß wir eine Vorabversion von Masic zum Testen in der Redaktion zur Verfügung hatten.

Neben der Tonhöhe und der Tondauer gibt es weitere Befehle in Masic, mit denen sich Musikstücke einfach programmieren lassen. Wie in anderen Programmiersprachen existieren zum Beispiel Sprungbefehle zur Programmierung von Endlosstücken, sowie Unterprogrammaufrufe, damit ein und dieselbe Notenfolge nicht mehrmals eingegeben werden muß. Nebenbei bemerkt, spart das Speicherplatz. Wie schon angedeutet, gibt es einen Befehl, der die absolute Länge einer Note in 50stel-Sekunden-Schritten festlegt. Dann existieren Befehle, die Halleffekte, Glissandisowie Tremolo-Effekte erzeugen. Da die komponierten Musikstücke auch in eigene Programme eingebunden werden können, gibt es Wege, ein Musikstück mit externen Ereignissen zu synchronisieren. Es ist zum Beispiel keine Schwierigkeit, in einem selbstgeschriebenen Spiel auf Knopfdruck ein Musikstück erklingen zu lassen.

Das so zusammengestellte Musikstück wird mit einer Tastenkombination auf Diskette gespeichert. Eine andere lädt ein bereits gesichertes Stück wieder in den Speicher. Da Masic ein Compiler ist, muß jedes Musikstück zuerst kompiliert werden, bevor man es sich anhören kann. Einige eingefleischte Computerbesitzer schrecken jetzt wahrscheinlich zurück, denn das Kompilieren von Programmen ist normalerweise eine zeitaufwendige und fehlerträchtige Angelegenheit. Nicht so bei Masic. Ein Knopfdruck genügt, um das Programm zu kompilieren, selten dauert dieser Vorgang länger als ein paar Sekunden. Einer weiterer Knopfdruck, und das Stück erschallt aus dem Lautsprecher eines angeschlossenen Fernsehers oder Monitors.

Mit Masic ist es aber auch möglich, eigenständig lauffähige Programme zu erzeugen, die dann vom DOS aus geladen werden und die Melodie abspielen. Dazu erteilt man kurz nach dem Kompilieren einen Speicherbefehl. Auf der Diskette, die das Musikstück aufnehmen soll, müssen zwei Files stehen, die im Lieferumfang des Programms enthalten sind. Diese Files stellen sozusagen das Runtime-Package von Masic dar. Beim Schreiben auf Diskette werden dann aus diesen Files die Routinen herausgeholt und mit dem jeweiligen Musikstück verbunden. Auf der Diskette steht jetzt ein lauffähiges Programm, das das Musikstück abspielt.

Compiliert auf Knopfdruck

Im geschickt aufgebauten Handbuch wird ausführlich erklärt, wie man derart erzeugte Programme in eigene Programme einbindet.

Masic ist ein gut durchdachtes Programm, mit dem sich Musikstücke mit sehr professionellem Sound erzeugen lassen. Was andere Musikprogramme dieser Art versäumt haben, ist hier eingebaut worden. Die an sich hervorragende Idee, eine eigene Programmiersprache zum Gestalten von Musikstücken zu verwenden, leidet etwas unter der Tatsache, daß Musik doch eine sehr emotionale Angelegenheit ist. Bei einem normalen Instrument wird jede Veränderung am Musikstück sofort hörbar. Bei Masic müssen erst verschiedenste Parameter eingeben werden, bis aus dem Computer Töne erklingen. Hat man jedoch erst einmal Erfahrungen im Schreiben von Musikstücken gesammelt, oder steht gar ein Instrument zur Verfügung, mit dem sich Tonfolgen ausprobieren lassen, ist Masic ein ideales Mittel, Musik für die Atari-Computer zu komponieren. Wer Masic kaufen möchte, sollte sich an folgende Adresse wenden: Verlag Rätz-Eberle, Melanchthonstr. 75/1, Postfach 1640, 7518 Bretten. Das Programm kostet 49 Mark. (hf)

DR. NIBBLE & crew







Pascal mit Assembler-Power

Es gehört zu den weitverbreiteten Vorurteilen, daß es auf den 8-Bit-Ataris keine leistungsfähigen Programmiersprachen gibt. Kyan-Pascal widerlegt dies!

Wer heute an der Schule oder an der Universität zum ersten Mal eine Programmiersprache kennenlernt, beginnt in der Regel mit Pascal. Auch in den Schulen fängt Pascal an, sich im Informatik-Unterricht durchzusetzen. Das hat auch gute Gründe. Pascal vereint große Flexibilität und Geschwindigkeit mit klarer Gliederung. Das ist neben der weitgehenden Standardisierung auch ein Grund, warum viele Software-Häuser ihre Programme in Pascal entwickeln.

Aber gerade was das Lernen einer Hochsprache angeht, hat Pascal die Nase vorne. Man lernt von Anfang an, problemorientiert zu denken und die Aufgabe erst völlig zu verstehen, bevor man ans eigentliche Programmieren geht. Das ist zwar am Anfang etwas frustrierend, erweist sich hinterher aber immer als hilfreich. Eingefleischte Pascal-Fans pflegen auf die Startschwierigkeiten zu antworten, daß man beim Laufenlernen auch mehr hingefallen als auf zwei Beinen gestanden sei. Aber das sei schließlich kein Argument, um sein ganzes Leben auf allen Vieren durch die Welt zu krabbeln. Wer will ihnen in diesem Punkt widersprechen?

Pascal hat für den Einsteiger aber noch weitere Vorteile, da sich viele Sprachen an Elemente aus Pascal anlehnen. Der spätere Umstieg auf »C« oder »Modula 2« bereitet keine Probleme, weil sich diese Sprachen sehr ähneln. Pascal ist also in keinem Fall eine Sackgasse.

Kyan contra Turbo?

Wenn man die Entwicklung und Verbreitung von Pascal betrachtet, findet man eine Art Generationenwechsel, der mit dem Erscheinen des legendären »Turbo-Pascal« zusammenhängt. Früher dominierte »UCSD-Pascal« die Szene, das sich noch stark an die Vorgaben des Pascal-Erfinders N. Wirth hielt. Doch mit der Entwicklung der Computer zeigte sich, daß man für den täglichen Einsatz mehr braucht, als man sich seinerzeit vorstellen konnte.

Mit dem Auftauchen von Turbo-Pascal der bekannten Firma Borland wandelte sich die Pascal-Szene. Mit Turbo-Pascal hat diese Sprache den Sprung aus dem Schatten der reinen Lehrsprache geschafft. Heute gibt es eigentlich keine Anwendung, die nicht auch in Pascal geschrieben werden kann. Für die PCs gibt es sogar Spiele, die in Pascal programmiert wurden.

Dem Atari-Besitzer kann es aber eigentlich gleich sein, denn es wird keine der beiden Pascal-Versionen für die 8-Bit-Ataris angeboten. Scheidet damit der Atari als Lerninstrument und Pascal als Programmiersprache auf dem Atari aus? Mitnichten!

```
 BIBO-Clock        Mo 15.12.1986      09:13:15
    Kyan Pascal
    Copyright (C) 1986 by
    Kyan Software, Inc.
    San Francisco, CA 94123

    Enter Command line or
    type "menu" for help:

% PC
pascal compiler
copyright 1986 by kyan software, inc.
usage:
[options] FileName [options]
options:
-d                  debug
-o FileName         output to file
-p                  progress report
-s                  do not assemble
>device             send errors to device

PC: █
```

So zeigt sich der Compiler auf dem Bildschirm

Seit fast zwei Jahren wird der schon auf dem Apple II erfolgreiche »Kyan-Pascal«-Compiler auch für den 8-Bit-Atari mit mindestens 64 KByte angeboten. Dabei handelt es sich nicht um eine abgemagerte Version des Apple-Originals, sondern um ein vollständiges und sehr leistungsfähiges System. Schon beim ersten Durchblättern des englischen Handbuchs sieht der Pascal-Kenner, daß Kyan-Pascal eine entscheidende Grundvoraussetzung für die sinnvolle Anwendung eines Pascal-Compilers erfüllt: Es beherrscht den Pascal-Sprachstandard voll und ganz. Was nützt schon ein Spar-Compiler, bei dem gerade die interessantesten Funktionen, die das Flair von Pascal ausmachen, fehlen?

Der Compiler verarbeitet den gesamten Umfang von ISO-Pascal, das weitgehend der ursprünglichen Definition durch Jensen/Wirth entspricht. Hier hat man also nicht die Rechnung ohne den Wirt(h) gemacht, wenn man sich auf Angaben über Standard-Pascal verläßt. Nebenbei sind auch Funktionen zur Stringbehandlung wie unter Turbo-Pascal implementiert. Was den Sprachumfang angeht, ist der Compiler also auf dem neuesten Stand der Dinge.

Komfortable Kommandos

Wie gestaltet sich nun der Umgang mit Kyan-Pascal? Der Compiler wird zusammen mit einem Text-Editor und dem Kommandointerpreter »KIX« für Atari DOS 2.5 geliefert. Durch dieses gut abgestimmte System verläuft die Arbeit mit Kyan-Pascal ähnlich wie unter den Betriebssystemen DOS XL, MS-DOS und UNIX. Viele Befehle von KIX sind den erwähnten Betriebssystemen sogar entliehen. Eine Funktion verleiht dem Ganzen ganz besonders den Hauch eines Großrechners. Bei den meisten Kommandos kann das Ausgabegerät mit dem Größer-Zeichen » > « bestimmt werden. Zum Beispiel gibt »LS *.* >P« das Directory auf dem Drucker aus.

Eleganter Editor

Wie bei allen Pascal-Systemen sind Compiler und Editor voneinander getrennte Programme, die erst über KIX geladen werden. Das Schreiben oder Verbessern eines Pascal-Programms beginnt also immer mit dem Laden des Bildschirmeditors »ED«. Beim Editieren stehen dem

Benutzer nicht nur Funktionen zum Löschen und Einfügen von Zeichen und Zeilen zur Verfügung. Auch Befehle zum Ausschneiden, Kopieren und Löschen von Textblöcken fehlen nicht. ED entpuppt sich beim Arbeiten als kleines, sehr komfortables Textverarbeitungssystem. Durch seinen vollständigen Befehlssatz und die hohe Arbeitsgeschwindigkeit ist es bestens zur Eingabe von Pascal-Programmtexten geeignet.

Inline-Assembler

Ist das Programm fertig eingegeben, verläßt man den Editor, um den Compiler zu starten. Das über 26 KByte lange Programm wird von KIX aus aufgerufen, zum Beispiel durch »PC HALLO.P«.

Durch die Trennung von Editor und Compiler gestaltet sich das Arbeiten öfter etwas kompliziert. Tritt nämlich ein Fehler auf, muß man wieder den Editor laden, um nach dem Fehler zu suchen, diesen verbessern und erneut mit dem Compilieren anfangen. Turbo-Pascal wurde übrigens auch dadurch bekannt, daß es Compiler und Editor gleichzeitig im Speicher hält. Bei langen Programmen, die fast immer kleine Bugs enthalten, dauert es daher manchmal etwas länger, bis man das Programm endlich testen kann. Das gehört zu den angesprochenen Problemen für Pascal-Neulinge. Die Arbeit lohnt sich aber.

Der Compiler wandelt das angegebene Pascal-Programm direkt in einen Assembler-Quelltext um, der anschließend automatisch vom dazugehörigen Makro-Assembler zum ausführbaren Programm assembliert wird.

Diese ungewöhnliche Lösung hat einige bestechende Vorteile gegenüber einer Direkt-Compilierung in Maschinensprache. Zum einen kann man dadurch ganze, vollständig in Assembler geschriebene Prozeduren einbinden. Zum anderen ist auch eine nachträgliche Optimierung des erzeugten Maschinencodes erlaubt. Zusätzlich kann man sogar genau wie bei Turbo-Pascal Assembler-Befehle direkt in das Pascal-Programm hineinschreiben. Der Assembler versteht nicht nur gängige Pseudo-Opcodes, sondern verarbeitet auch einfache Makros.

Besonders an dieser Funktion zeigt sich das durchdachte Konzept des Compilers. Er bietet nicht nur den Komfort von Pascal, sondern erlaubt auch sehr schnelle und variable Programme, die auch eingefleischte Assembler-Freaks befriedigen. Es ist sogar möglich, schwer umzusetzende Programmteile in Pascal zu schreiben, um den Assembler-Teil in ein anderes Assembler-Programm zu übernehmen. Das bietet sich besonders bei aufwendigen Berechnungen an, die in Assembler nur sehr schwierig zu programmieren sind.

Die fertigen Maschinensprach-Programme sind sehr kurz, da die Pascal-Laufzeitbibliothek, die die wichtigsten Standardprozeduren enthält und auch die Abarbeitung des Programms überwacht, nicht mit eingebunden, sondern erst bei Bedarf von Diskette geladen wird. Dies hat den großen Vorteil, daß man auf der Diskette eine Menge compilierte Programme unterbringen kann. Wer das häufige Diskettenwechseln von anderen Compilern her kennt, weiß diese Tatsache zu schätzen.

Bei den Diskettenzugriffen ist es den Entwicklern wiederum gelungen, das System sehr anwenderfreundlich zu machen. Alle Dateien werden von KIX zunächst auf der RAM-Disk (sofern vorhanden) gesucht. Das bedeutet, daß viele Teile nur ein einziges Mal geladen und beim nächsten Zugriff direkt aus dem Speicher geholt werden. Durch diesen einfachen Trick fällt bei genügendem Speicherausbau die zusätzliche Ladezeit überhaupt nicht ins Gewicht. Mit einem Atari 130 XE hat man also von Anfang an ein sehr schnelles System. Erfahrenen Programmierern wird es allerdings auch nicht schwerfallen, die Standardbibliothek direkt in das fertige Programm zu übernehmen.

Kurz und schnell

Wie steht es nun mit der Geschwindigkeit des ausführbaren Programms? In der Bedienungsanleitung wird versprochen, daß es in etwa die gleiche Geschwindigkeit wie ein unter Turbo-Pascal geschriebenes Programm auf einem mit 4 MHz laufenden CP/M-Rechner erreicht. Somit ist es beispielsweise mehr als doppelt so schnell wie auf einem Commodore 128 im CP/M-Modus. Dieses Versprechen wird nach unseren Tests auch gehalten. Das heißt in der Praxis, daß Kyan-Pascal neben der C-ähnlichen Sprache »ACTION!« die schnellste höhere Programmiersprache auf dem Atari ist. Sie ist zwar nicht so schnell wie reiner Assembler-Code, doch dafür wesentlich einfacher einzugeben und zu debuggen als Assembler.

Nebenbei soll noch lobend erwähnt werden, daß Kyan in der jetzt vorliegenden Version sämtliche Schwächen der ersten Ausführung beseitigt hat.

Kyan-Pascal ist eines der Produkte, denen man ihre klare Konzeption sehr deutlich anmerkt. Es ist klar durchdacht und sehr anwenderfreundlich. Man erhält für knapp 250 Mark ein Paket, das man auch zum Erlernen der Programmiersprache Pascal benutzen kann, da sämtliche gegebenen Standards erfüllt werden. Dazu trägt auch das ausgezeichnete Handbuch bei, das 170 der knapp 250 Seiten einer ausführlichen Einführung in die Programmiersprache widmet. Das einzige Manko ist, daß die Dokumentation leider in Englisch geschrieben ist. Wenn man sich hier nicht ganz sicher fühlt, empfiehlt sich ein deutsches Buch als Unterstützung zum Pascal-Lernen.

Eine der Besonderheiten an Kyan-Pascal ist, daß der fortgeschrittene Programmierer auch die Kommandoumgebung KIX und den mitgelieferten Assembler erhält. Dieses starke Paar unterstützt auch den Benutzer, der nicht nur Pascal programmieren möchte. Wer also nach einem relativ billigen Pascal-System sucht, ist mit einem Atari 130 XE und Kyan-Pascal bestens bedient. Es ist eine zukunftssichere Investition.

(Julian Reschke/gn)

Bezugsquelle: Compy Shop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr

KIX-Kommandos

CD	CHANGE DEVICE	Standardgerät (z.B. »D2:«) einstellen
PWD	PRINT WORKING DEVICE	Name des Standardgeräts ausgeben
LS	LIST DIRECTORY	Directory ausgeben
CAT	CONCATENATE	Dateiinhalt ausgeben
CP	COPY	Datei kopieren
MV	MOVE	Datei bewegen (kopieren und Quelle löschen)
RM	REMOVE	Datei löschen
CHMOD	CHANGE PROTECTION MODE	Dateistatus ändern (»LOCK«...)
FORMAT		Diskette formatieren
SD	SCREEN DUMP	Bildschirminhalt ausdrucken
DOS		Rückkehr in DOS 2.5
ED	EDITOR	Editor aufrufen
PC	PASCAL-COMPILER	Pascal-Compiler starten
AS	ASSEMBLER	Assembler starten

Der Star unter den Textverarbeitungen

Einfache Bedienung und schnelle Verarbeitung sind die Hauptmerkmale des Programms »StarTexter«. Es ist eine der besten Textverarbeitungen für den Atari XL.

Für Ataris 8-Bitter gibt es schon lange Zeit Textverarbeitungsprogramme. Als Beispiele seien hier der »Atari Schreiber«, »Atari-Wordprocessor«, »Letter Perfect« oder »Atext« genannt. In die Reihe der Altmeister reiht sich der »StarTexter« aus der »Mister Micro«-Serie des Sybex-Verlages ein. Dieses Programm vereint flexible Textgestaltung mit einfacher Bedienung. Ein Programm, auf das viele gewartet haben.

Nach dem Start von StarTexter öffnet sich erst einmal lautstark ein Schott auf dem Bildschirm und gibt den Blick auf das Textfeld frei. Diesen Scherz hat man von der C64-Version übernommen. Der rasante Beginn kündigt aber schon an, was den Benutzer beim Arbeiten mit dem StarTexter erwartet: ein spielerisch leicht zu bedienendes Programm.

Das Textfenster, in dem gearbeitet wird, umfaßt 21 Zeilen mit je 40 Zeichen. Am oberen Bildschirmrand befindet sich eine Informationszeile, die auf einen Blick die wichtigsten Parameter anzeigt. Durch Scrolling bewegt sich das Textfenster auf der Arbeitsfläche recht komfortabel senkrecht und waagrecht. Der Text darf ohne weiteres breiter sein, als er auf einmal auf dem Bildschirm dargestellt wird. Beim Scrollen im Text ist der StarTexter sehr schnell, so daß auch bei großen und umfangreichen Dokumenten keine merkliche Verzögerung eintritt. Das ist für alle wichtig, die viel mit dem Programm arbeiten wollen.

Die aktuelle Cursorposition im Text wird in der unteren Bildschirmzeile angezeigt. Das Arbeitsblatt kann man nach Lust und Laune mit Text füllen, ohne daß der Benutzer auf Zeilenlänge oder Absätze achten muß. Eine deutsche Tastaturbelegung mit Umlauten ist selbstverständlich. Korrekturen sind durch Überschreiben, Löschen oder Einfügen von Zeichen leicht durchführbar.

Der StarTexter kennt zwei Schreibmodi. Entweder überschreibt er alle Zeichen, die sich eventuell in der Zeile befinden, oder er schiebt sie erst zur Seite, bevor der neue Buchstabe gesetzt wird. Die Umschaltung zwischen Überschreiben und Einfügen erfolgt bequem per Tastendruck.

Auch Blockoperationen bereiten diesem Programm keine Probleme. Ganze Textblöcke können kopiert, gelöscht oder verschoben werden. Hier bietet der StarTexter alle Standardfunktionen, die das Schreiben angenehm machen. Beim Verfassen von langen Texten kommt es, daß man das Geschriebene umstrukturieren möchte. Mit diesem Programm ist das kein Problem.

Ist der Text komplett eingegeben, kann man sich mit dem Aussehen des Schriftstücks, der Formatierung, befassen. Durch Druck auf die <ESC>-Taste wird der sogenannte »Control«-Modus eingeschaltet. Mit den Funktionstasten <OPTION>, <SELECT> und <START> werden dann die Menüs aufgerufen. Die einfache und einprägsame Menüführung macht den StarTexter so anwenderfreundlich, daß man keine große Erfahrung mit anderen Textverarbeitungsprogrammen braucht, um schnell mit diesem Programm zurecht zu kommen.

Durch die <START>-Taste gelangt man im Control-Modus in das Formatierungs- und Druckausgabe-Menü. Hier wird der Text des Arbeitsblattes automatisch auf das gewünschte Format gebracht und dann ausgedruckt. Zur Formatierung dienen die mit dem OPTION-Menü eingestellten Parameter für Zeilen- und Seitenlänge sowie für Einrückung von Absätzen. Die Formatierung kann sowohl vorher auf dem Bildschirm als auch direkt bei der Druckausgabe erfolgen. Die vorherige Formatierung hat natürlich den Vorteil, daß man sich schon am Monitor einen Eindruck vom Aussehen des Texts verschafft.

Im Zusammenhang mit der Formatierung zeigt der StarTexter aber seine größte Schwäche. Normalerweise passen bis zu 20000 Zeichen in den Textspeicher. Diese Zahl ergibt sich aus der maximalen Textbreite von 80





So präsentiert sich der StarTexter auf dem Bildschirm: Mit deutschen Umlauten und übersichtlichen Menüs

Zeichen und der maximalen Zeilenzahl von 250 Zeilen. Wenn man den Text aber auf dem Bildschirm formatiert, geht der StarTexter recht großzügig mit dem Speicherplatz um. Jede Zeile wird nämlich im Textspeicher und in einem externen Speichergerät mit einer Länge von 80 Zeichen berücksichtigt, auch wenn die Zeilenlänge zum Beispiel auf 60 reduziert wurde. Dieser ungenutzte Teil des Speicherplatzes steht aber nicht in Form von zusätzlichen Zeilen zur Verfügung. Der nicht genutzte Teil wird durch Leerzeichen aufgefüllt. Wenn der Text auf 60 Zeichen pro Zeile formatiert ist, können nur 250 x 60 = 15000 Zeichen statt der eigentlich nutzbaren 20000 Zeichen verwendet werden. Diese Platzverschwendung ist sehr störend, selbst wenn 250 Zeilen eine Menge Text sind.

Platzverschwendung

Die Programmautoren sind dabei wohl den für sie einfachsten Weg gegangen. Für den passionierten Vielschreiber ist das aber sehr ärgerlich, da man nicht in allen Fällen mit der vollen Zeilenbreite arbeiten will. Längere Texte muß man also zwangsläufig auf mehrere Dateien verteilen, wodurch das Finden des Anschlusses teilweise sehr schwierig ist. Wenn nämlich mitten im Satz der Textspeicher voll ist, muß man entweder die Folgedatei an exakt derselben Stelle fortsetzen oder den Text bis zum letzten Absatz löschen. Das scheint der Preis für die hohe Verarbeitungsgeschwindigkeit zu sein. Der einzige Weg, um den Speicher voll zu nutzen, ist, nur zum Drucken zu formatieren.

Mit der <SELECT>-Taste ruft man das Disketten-Menü auf, das eine Vielzahl von Diskettenoperationen wie Laden und Speichern von Texten, Formatieren von Disketten, Umbenennen von Dateien, Laden eines anderen Zeichensatzes und Anzeigen des Directory erlaubt.

Im <OPTION>-Menü befinden sich umfangreiche Einstellungen für die Textdarstellung, wie das Setzen von Randbegrenzungen, Seitenlänge, Wortumbruch, Blocksatz, Zeilenabstand, Zeilenlänge und Position der Kopf- und Fußzeile. Die farbliche Darstellungsart von Bildschirmrand, Cursorfarbe und Blockmarkierung läßt sich ebenfalls sehr vielfältig manipulieren.

Bevor der Text gedruckt wird, kann man ihn sich durch Druck auf die HELP-Taste in 80-Zeichen-Darstellung auf dem Bildschirm ansehen. Aufgrund der beschränkten Auflösung der Atari-XL/XE-Computer ist die Lesbarkeit natürlich nicht besonders hoch, aber um sich einen Überblick über das Aussehen des Schriftstückes zu verschaffen, reicht sie allemal aus. Sinn macht diese Funktion natürlich nur, wenn man den Text auch im Speicher formatiert. Wählt man aus Platzgründen nur die Druckformatierung, nutzt diese sonst sehr nützliche Funktion gar nichts.

Bevor man mit StarTexter etwas zu Papier bringen will, sollte man auf jeden Fall das Installationsprogramm »Install.Bas« starten. Sofern man nicht einen voreingestellten Atari-Drucker 1025 oder 1029 oder einen Epson-kompatiblen Drucker verwendet, kann man mit dem Installationsprogramm seinen Drucker individuell an den StarTexter anpassen. Während bei anderen Programmen oft Probleme mit dem Druckeranschluß auftreten, haben sich die Autoren des StarTexters bei diesem Punkt sehr viel Mühe gegeben. So kann man zum Beispiel einen Drucker mit Centronics-Schnittstelle über die Joystickports anschließen, wenn kein Centronics-Interface zur Verfügung steht! Dadurch kommt der Benutzer auch in den Genuß eines Centronics-Interfaces, wie es zum Ansteuern der meisten Drucker nötig ist.

Als kleines »Bonbon« gibt es noch die Funktion, in der Textverarbeitung zu rechnen! Die Programmierer nutzen hierbei den eingebauten Basic-Interpreter aus. Die zu berechnenden Ausdrücke werden dem Basic-Interpreter als Basic-Zeile »verkauft«, die dann berechnet und an StarTexter ausgegeben werden. Diese Funktion lohnt sich besonders für denjenigen, der mit seinem Computer Rechnungen schreiben will. Der zusätzliche Taschenrechner hat beim StarTexter ausgedient. Aber auch für Referate in Mathematik oder Physik kann man die Rechenfunktion gut gebrauchen.

Als deutsches Produkt hat der StarTexter wie erwähnt schon von vorn herein einen deutschen Zeichensatz und die deutsche Tastaturbelegung. Nur gibt es fast immer einen Anlaß, ein bestimmtes Zeichen drucken zu wollen, das nicht im Zeichensatz enthalten ist. Besonders bei fremdsprachigen Texten tritt dieses Problem häufig auf. Der StarTexter weiß hier Abhilfe.

Mit dem ebenfalls auf der Diskette enthaltenen Basic-Programm »StarFont« entwirft man eigene Zeichensätze für den StarTexter. Drei fertige Zeichensätze werden übrigens gleich mitgeliefert. Wenn man es wünscht, werden die neuen Zeichen auf dem Bildschirm dargestellt. Ein Ausdruck von Texten mit verschiedenen Zeichensätzen ist natürlich ebenfalls möglich. Das mühsam gezeichnete Telefon-Symbol oder die griechischen Buchstaben erscheinen beim StarTexter also garantiert auch auf dem Drucker und nicht wie bei anderen Programmen nur auf dem Bildschirm.

Eine Funktion vermißt man bei StarTexter leider. Er beherrscht keine Serienbriefe, so daß er zum Beispiel für die Vereinsverwaltung nur bedingt geeignet ist. Man muß bei jedem Brief die Adresse, den Namen und die Anrede von Hand ändern. Bei drei identischen Briefen stört das noch nicht, aber bei zehn oder mehr geht diese Funktion sehr ab. Hier zeigt sich der Nachteil, daß Sybex im Gegensatz zur C64-Version keine passende Dateiverwaltung zum StarTexter anbietet. Wer allerdings selten den gleichen Brief an viele Leute verschicken möchte, wird die Serienbrief-Funktion kaum vermissen.

Ein großes Lob gebührt dem Programm wegen der Dokumentation. Das Handbuch ist ausführlich und verständlich geschrieben. Es hilft dem Einsteiger, sich schnell auf dem StarTexter einzuarbeiten, so daß die ersten Texte nicht lange auf sich warten lassen. Kleine Beispiele im Handbuch dienen jeweils zur Verdeutlichung der erklärten Funktion.

Gutes Handbuch

Die Anleitung hilft aber nicht nur dem Neuling. Im Anhang befindet sich eine Übersicht aller Funktionen mit einer Kurzerklärung. Die Referenzkarte hat sich beim täglichen Einsatz bewährt, da man nicht erst im Handbuch nachschlagen muß. Auch wer viel mit einem Programm arbeitet, kennt nur die wichtigsten Standard-Befehle auswendig. Die sonstigen Fähigkeiten des Programms beziehungsweise wie man eine Zusatzfunktion korrekt aufruft, vergißt man schnell. Außerdem findet man sich auch schnell wieder zurecht, wenn längere Zeit nicht mit dem Programm gearbeitet wurde.

Im großen und ganzen erhält man für den Preis von 64 Mark ein leistungsfähiges Programm, das für die meisten Heimanwendungen vollauf ausreicht. Die Bedienung ist durch das ausführliche Handbuch und dank der ausgefeilten Menütechnik einfach und schnell zu erlernen. Der StarTexter ist voll und ganz auf den Heimbereich zugeschnitten und empfiehlt sich jedem, der eine unkomplizierte Textverarbeitung sucht.

(H.D. Jankowski/Julian Reschke/gn)

Atari verbindlich: XL-ST-Interface

Besitzen Sie neben dem Atari XL auch seinen großen Bruder, einen Atari ST, und möchten Programme und Dateien übertragen? Hier ist die Lösung: ein XL-ST-Interface!

Der Aufwand für ein Interface zwischen den beiden Computern ist verhältnismäßig gering. Sie benötigen einen XL, einen ST (oder einen 400er/800er), eine kleine Schaltung und ein Programm im ST, welches alles verwaltet. Sie erhalten dann einen aufgerüsteten XL mit zwei Diskettenlaufwerken und Drucker, die vom ST emuliert werden. Gleichzeitig können Sie beliebige Dateien oder auch den Inhalt ganzer Disketten vom XL zum ST und umgekehrt übertragen. Durch Änderung des Programmes kann der ST den XL steuern und als Peripheriegerät benutzen.

Die Hardware

Die Schaltung wird zur Pegelwandlung zwischen der RS232-Schnittstelle und dem XL-Systembus benötigt, da der ST mit 12 Volt und der XL-Bus mit TTL-Pegel (also 5 Volt) arbeitet. Es gibt hierfür einige Schaltungsvarianten. Da keine geeignete Spannungsversorgung bei den Computern vorhanden ist, haben wir eine Schaltung gewählt, die aus den am XL-Bus anliegenden 5 Volt selbständig die 12 Volt für den ST herstellt. Das Kernstück dieser Schaltung ist ein IC mit der Bezeichnung MAX232. Auf Seite 150 sehen Sie den recht einfachen Schaltplan. Das Bild rechts zeigt die fertig auf Lochrasterplatte aufgebaute Schaltung. Außer dem MAX232 benötigen Sie noch vier Elektrolyt-Kondensatoren. Beim Zusammenbau müssen Sie unbedingt auf die richtige Polung der Kondensatoren achten.

Die Software

Obwohl das Programm in GFA-Basic geschrieben ist, arbeitet es schneller als die wirklichen Diskettenlaufwerke des XL. Lediglich für die Routine zur Prüfsummenbildung wird ein Maschinenprogramm eingesetzt, das jedoch aus nur 24 Byte besteht. Außerdem läuft das gesamte Programm vollständig unter GEM, wodurch die Bedienung sehr einfach wird.

Nach dem Starten des Programmes erhält man außer der Menüleiste vier Fenster, je eines für die zwei virtuellen Diskettenlaufwerke, eines für den Drucker und ein Statusfenster. Die Laufwerke sind wahlweise ausschaltbar und haben dann keinerlei Einfluß auf die Busfunktionen des XL. Somit kann man zum Beispiel ein 1050-Laufwerk problemlos mitbenutzen, wobei Sie durch Ausschalten des echten und Einschalten des virtuellen Laufwerks hin- und herschalten.

Die virtuellen Laufwerke arbeiten zunächst einmal genauso wie die echten. Zusätzlich haben Sie die Wahl, die Floppies von 128 auf 256 Byte pro Sektor umzuschalten, wodurch sich die Diskettenkapazität verdoppelt. Ein käufliches Laufwerk mit 256 Byte pro Sektor hat es zwar hierzulande nie gegeben, das DOS II des XL unterstützt jedoch ein solches Gerät.

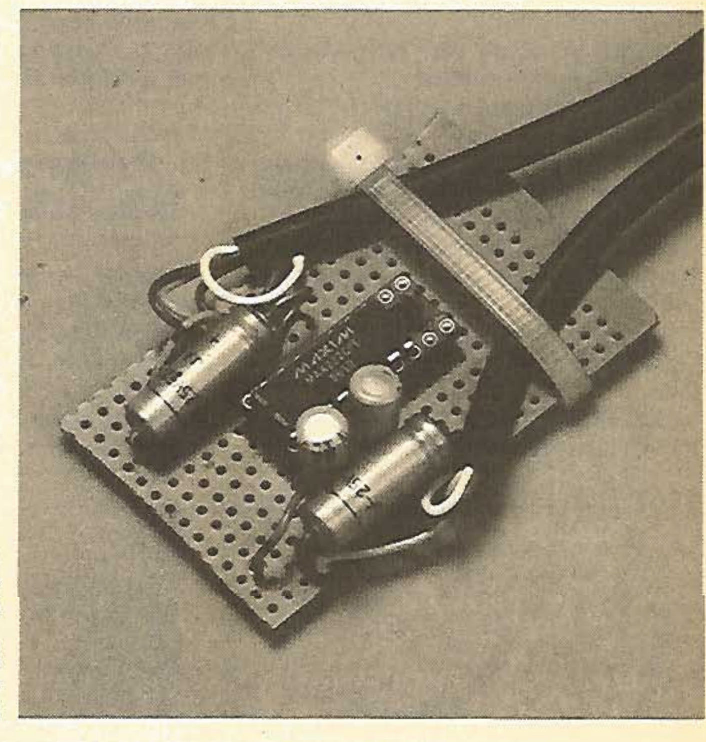
Sie können nun beliebige Dateien von XL- auf ST-Format und umgekehrt übertragen, wodurch zum Beispiel alte Texte oder Adreßdaten im XL für den ST verfügbar werden. Auch ganze Disketteninhalte können im ST oder auf 3½-Zoll-Diskette gespeichert und schnell wieder geladen werden, was sich für System- oder Compilerdisketten anbietet, da man natürlich den XL auch vom ST aus booten kann.

Das Menü des Programmes bietet folgende Auswahlpunkte:

Der erste Menüpunkt »Desk« enthält:

Atari 800 Interface gibt eine Programminformation aus.

Außerdem stehen die sechs Accessories zur Verfügung.



Auf einer kleinen Lochrasterplatine findet das Interface spielend Platz, man braucht nicht einmal ein Lötkünstler zu sein

Die folgenden zwei Menüs »Disk 1« und »Disk 2« beinhalten die Auswahlpunkte:

Directory zeigt das aktuelle Inhaltsverzeichnis und den verbliebenen Speicherplatz an

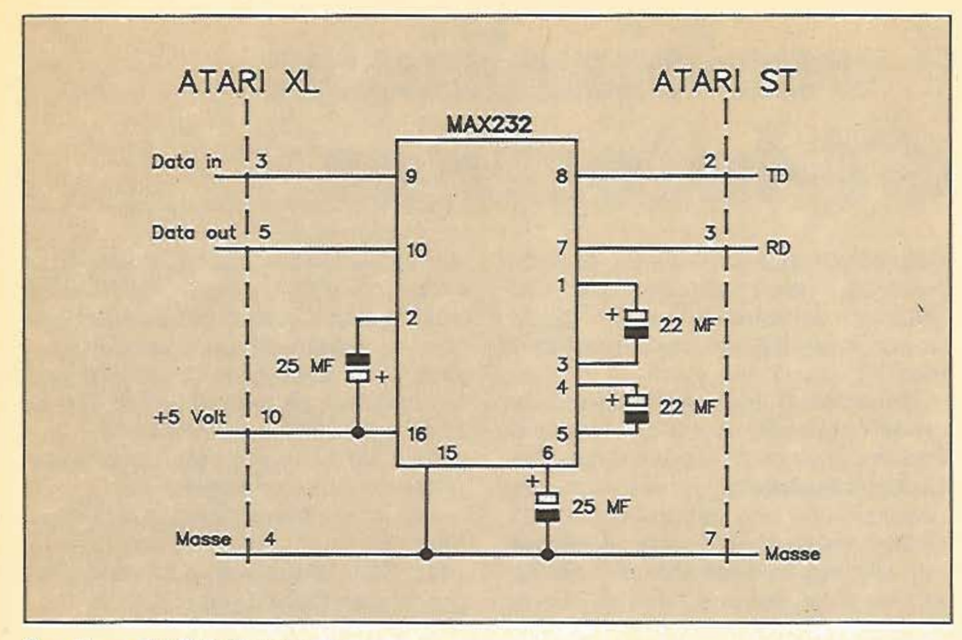
Save Disk speichert den Disketteninhalt auf ST-Diskette (nur die belegten Sektoren)

Load Disk lädt eine mit »Save Disk« gespeicherte Diskette

Format SD formatiert die Diskette im Single-Density-Format (DOS II)

Format DD formatiert im Double-Density-Format (DOS III)

Set Aktivierung/Desaktivierung der Diskette



Das Herz der Schaltung: IC MAX232

Das brauchen Sie für diese Schaltung:

IC:	MAX232
Kondensatoren:	2x 22µF
	2x 25µF
Stecker:	25pol D-Sub (weiblich)
	XL-System-bus-Stecker

Können Sie keinen Systembus-Stecker auftreiben, benutzen Sie einzelne Steckschuhe, die auf diese Stifte passen. Sie sind in jedem Elektronik-Laden erhältlich. Natürlich können Sie das Verbindungskabel auch im Atari XL von innen an den Systemport anlöten.

Zum Schluß benötigen Sie noch ein Stück Kabel, mindestens vieradrig, und ein kleines Stück Lochraster-Platine.

Im Menü »Settings« sind folgende Punkte enthalten:

128 BpS Auswahl von 128 Byte pro Sektor Diskettenformat
256 BpS Auswahl von 256 Byte pro Sektor
show Sector# Ein-/Ausschalten der ständigen Anzeige der Nummer des momentan bearbeiteten Sektors
Clearw Disk 1, Clearw Disk 2, Clearw Status löschen das angegebene Fenster

Unter dem Menüpunkt »Ende« steht **Quit**, womit man das Programm beendet.

Das Programm beginnt mit einigen Initialisierungen: Menü lesen und anzeigen, Fenster benennen und öffnen, Schnittstelle auf 19200 Baud einstellen, womit der serielle Bus des XL läuft, und den RS232-Puffer leeren.

Programm nach Wunsch erweitern

Danach wartet das Programm in einer Schleife auf den Empfang eines seriellen Byte oder auf die Wahl eines Menüpunktes. Ist ein Byte angekommen, testet eine Routine, ob es den Drucker oder eine aktivierte Diskettenstation anspricht. Wenn nicht, so wird es ignoriert, andernfalls liest das Programm den Rest des zu erwartenden Kommandoblocks und führt die geforderte Funktion aus. Bei eventuellen Prüfsummenfehlern wird eine entsprechende Fehlermeldung im Statusfenster ausgegeben, wobei die erste ausgegebene Zahl die selbsterrechnete und die zweite Zahl die empfangene Prüfsumme darstellt.

Um Platz zu sparen, haben wir auf die Drucker-, Kopier- und Directory-Routinen verzichtet. Dennoch ist diese Version lauffähig, so daß Sie das Programm nach Ihren eigenen Wünschen erweitern können.

(Stefan Dittrich/kl)

```
' *** ATARI ST-ATARI 800-
Interface  1/87
S.A.Dittrich ***
Cls
Deffn Cl=(Sec%-25)/8
Deffn Sec=8*Cl%+25
Deftext 1,0,0,4
Dim Com%(5),Me$(60),D$(1,1050)
,Df%(1),D%(1),Su%(1,1050)
Open "",#1,"AUX:"
Null$=String$(128,0)
Ff$=String$(256,255)
Len%=128
@Initall
Restore
For I%=0 To 40
  Read Me$(I%)
  Exit If Me$(I%)="***"
Next I%
Me$(I%)=""
Me$(I%+1)=""
Data Desk , ATARI 800
Interface,----------------
------,1,2,3,4,5,6,""
Data Disk 1 ,  Directory,
Save Disk,  Load Disk,
Format SD,  Format DD,---
----------,  Set,""
Data Disk 2 ,  Directory,
Save Disk,  Load Disk,
Format SD,  Format DD,---
----------,  Set,""
Data Settings ,  128 BpS,
256 BpS,  show Sector# ,-
----------------
Data   Clearw Disk 1,  Clearw
Disk 2,  Clearw Status,""
Data Ende, * Quit * ,***
Menu Me$()
Closew 1
Titlew 1,"Disk 1"
Infow 1,"-"
Closew 2
Titlew 2,"Drucker nicht
implementiert"
Infow 2,"Vollständige Version
auf Leserservice-Disk!"
Closew 3
Titlew 3,"Disk 2"
Infow 3,"-"
Closew 4
Titlew 4,"Status"
For I%=1 To 4
  Openw I%
  Cls
Next I%
On Menu  Gosub Menue
D%=Xbios(15,0,0,-1,-1,-1,-1)
D%(1)=1
Sn%=0
Menu 26,1
P%=1
Sh%=1
Menu 47,1
@Dinfo2
@Setmp
Go1:
Showm
While Bios(1,1)=-1
  B%=Inp(1)
  On Menu
Wend
Go:
Showm
While Bios(1,1)<>-1
  On Menu
Wend
Hidem
Go2:
C%=Inp(1)
If (C%<>64 Or P%=0) And
(C%<>Asc("1") Or D%(0)=0)
 And (C%<>Asc("2") Or
D%(1)=0)
  Goto Go
Endif
S%=C%
For I%=1 To 3
  B%=Inp(1)
  Com%(I%)=B%
  S%=S%+B%
  S1%=(S% And 255)+Int(S%/256)
Next I%
B%=Inp(1)
If B%<>S1%
  Openw 4
  Print " Invalid Command-
Frame !"
  Goto Go1
Endif
If C%=Asc("1") Or C%=Asc("2")
  ' *  Disk-Routinen  *
  D%=C%-Asc("0")-1
```

```
  Out 1,Asc("A")
  Sec%=Com%(2)+Com%(3)*256
  If Com%(1)=Asc("R")
    ' *  Read Sektor  *
    Out 1,Asc("A")
    If Sn%
      Openw D%*2+1
      Print At(1,1);"Sending
Sector $";Hex$(Sec%);" ("
;Sec%;")      "
    Endif
    Print #1,D$(D%,Sec%)+
Chr$(Su%(D%,Sec%))+"C"
    Goto Go
  Endif
  If Com%(1)=Asc("P") Or
Com%(1)=Asc("W")
    ' *  Put/ Write Sektor *
    Out 1,Asc("A")
    If Sec%<4
      D$(D%,Sec%)=Input$(128,
#1)
    Else
      D$(D%,Sec%)=Input$(Len%,
#1)
    Endif
    B%=Inp(1)
    Out 1,Asc("A")
    If Sn%
      Openw D%*2+1
      Print At(1,1);"Getting
Sector $";Hex$(Sec%);" ("
;Sec%;")      "
    Endif
    @Dosum(Sec%)
    If S1%<>B%
      Openw 4
      Print " PUT: Checksum-
Error in $";Hex$(Sec%);"
: $";Hex$(S1%);"<>$";
Hex$(B%)
      Out 1,Asc("E")
      Goto Go1
    Endif
    Out 1,Asc("C")
    Goto Go
  Endif
  If Com%(1)=Asc("S")
    ' *  Disk-Status  *
    Openw D%*2+1
    If Len%=128
      Restore Dstat1
    Else
      Restore Dstat2
    Endif
    For I%=1 To 6
      Read B%
      Out 1,B%
    Next I%
    Goto Go
    Dstat1:
    Data 67,16,0,1,0,17
    Dstat2:
    Data 67,48,0,1,0,49
  Endif
  If Com%(1)=Asc("!") Or
Com%(1)=      34
    ' *  Format Disk SD oder
DD  *
    Out 1,Asc("A")
    Df%(D%)=Com%(1)-32
    @Dinfo
    Print " Formatierung.."
    @Init
    Print #1,String$(Len%+1,
255)
    Out 1,Asc("C")
    Goto Go2
  Endif
  Openw 4
  Print "Unknown Command : ";
Hex$(Com%(1))'"(";
Chr$(Com%(1));")"
  Goto Go1
Endif
Goto Go
' *  Prozeduren  *
Procedure Menue
  D%=1+(Menu(0)<19)
  On Menu(0) Gosub Info
  On Menu(0)-10 Gosub Dir,Sav,
Loa,Fmtsd,Fmtdd,X,     Setd
  On Menu(0)-19 Gosub Dir,Sav,
Loa,Fmtsd,Fmtdd,X,     Setd
  On Menu(0)-28 Gosub Set128,
Set256,Shownr,X,Clw,Clw,
Clw,Clw
  Menu Off
  If Menu(0)=38
    Showm
    Menu Kill
    Deftext 1,0,0,13
    Stop
    Menu Me$()
    Deftext 1,0,0,4
    On Menu  Gosub Menue
  Endif
Return
Procedure Info
  Alert 0,"  ATARI 800-
Interface  :    1/87
S.Dittrich   :",1," OK ",
Dummy%
Return
Procedure Clw
  Openw Menu(0)-31
  Cls
Return
Procedure Setmp
  Restore Mprog
  M$=""
  For I%=1 To 24
    Read A$  .
    M$=M$+Chr$(Val("&H"+A$))
  Next I%
  Mprog:
  Data 2a,6f,00,04,42,85,52,
45,3a,2f,00,08,42,80,12,
1d,d1,01,51,cd,ff,fa,4e,
75
Return
' *  Disk-Prozeduren  *
Procedure Set128
  If Len%=256
    @Sure
    If Sure%=2
      Null$=String$(128,0)
      Menu 28,1
      Menu 29,0
      Len%=128
      @Initall
    Endif
  Endif
Return
Procedure Set256
  If Len%=128
    @Sure
    If Sure%=2
      Null$=String$(256,0)
      Menu 28,0
      Menu 29,1
      Len%=256
      @Initall
    Endif
  Endif
Return
Procedure Sure
  Alert 1," Größenveränderung
löscht:      alle Daten !:
",1," Stop : OK ",      Sure%
  If Sure%=2
    Arrayfill Df%(),0
    @Dinfo2
  Endif
Return
Procedure Shownr
  If Sn%
    Sn%=0
  Else
    Sn%=1
  Endif
  Menu 49,Sn%
Return
'
Procedure Dosum(O%)
  If O%<4
    L%=128
  Else
    L%=Len%
  Endif
  M%=Varptr(M$)
  S1%=C:M%(L:Varptr(D$(D%,O%))
,L%)
  Su%(D%,O%)=S1%
Return
Procedure Dir
  ' * Directory anzeigen
  ' * ist nicht implementiert
Return
Procedure Initall
  For D%=0 To 1
    @Init
  Next D%
Return
Procedure Init
  For J%=1 To 3
    D$(D%,J%)=String$(128,0)
    Su%(D%,J%)=0
  Next J%
  For J%=4 To 1050
    D$(D%,J%)=String$(Len%,0)
    Su%(D%,J%)=0
  Next J%
  Openw D%*2+1
  Cls
Return
Procedure Sav
  If Df%(D%)
    @File
    If F$<>""
      Defmouse 2
      For I%=1050 Downto 1
        Exit If D$(D%,I%)
<>String$(Len%,0)
      Next I%
      Open "R",#2,F$,Len%+1
      Field #2,Len% As A$,1
As     Su$
      A1$=Str$(I%)+"D"+
Str$(Df%(D%))+"L"+
Str$(Len%)
      Su$=" "
      Lset A$=A1$
      Put #2,1
      For J%=1 To I%
        Lset A$=D$(D%,J%)
        Su$=Chr$(Su%(D%,J%))
        Put #2,J%+1
      Next J%
      Close #2
      Defmouse 0
    Endif
  Endif
Return
Procedure Loa
  W=1
  If Df%(D%)
    Alert 1," Laden löscht
den :   alten Inhalt:",1,
" OK : Cancel ",      W
  Endif
  If W=1
    @File
    If F$<>""
      Defmouse 2
      @Init
      Open "I",#2,F$
      A$=Input$(129,#2)
      I%=Val(A$)
      Df%(D%)=Val(Mid$(A$,
Instr(A$,"D")+1,4))
      If Len%=Val(Mid$(A$,
Instr(A$,"L")+1,4))
        If Len%=256
          A$=Input$(128,#2)
        Endif
        For Sec%=1 To I%
          D$(D%,Sec%)=
Input$(Len%,#2)
          Su%(D%,Sec%)=
Asc(Input$(1,#2))
        Next Sec%
        For Sec%=1 To 3
          D$(D%,Sec%)=
Left$(D$(D%,Sec%),128)
        Next Sec%
      Else
        Alert 1," Falsche
Sektor-Länge :",1," Stop
",Dummy%
      Endif
      Close #2
      @Dinfo
    Endif
    Defmouse 0
    @Dir
  Endif
Return
Procedure File
  Fileselect "\*.dsk","",F$
  If Instr(F$,".")=0 And F$<>"
"
    F$=F$+".dsk"
  Endif
Return
Procedure Setd
  Wi=2*D%+1
  Openw Wi
  If D%(D%)
    D%(D%)=0
```

Verwaltungsprogramm im GFA-Basic für das XL-ST-Interface

```
      Menu 17+9*D%,0
      Infow Wi," Inaktiv"
    Else
      D%(D%)=1
      Menu 17+9*D%,1
      @Dinfo
    Endif
Return
Procedure Fmtsd
    @Fmtsure
    If Sure%=1
      Mid$(D$(D%,360),1,5)=
Chr$(2)+Chr$(&HC3)+
Chr$(2)+Chr$(&HC3)+
Chr$(2)
      Mid$(D$(D%,360),11,1)=
Chr$(15)
      Mid$(D$(D%,360),12,89)=
String$(89,255)
      Mid$(D$(D%,360),56,2)=
Chr$(0)+Chr$(&H7F)
      Su%(D%,360)=28
      Df%(D%)=1
      @Dinfo
    Endif
Return
Procedure Fmtdd
    @Fmtsure
    If Sure%=1
      Mid$(D$(D%,16),15,2)=
Chr$(&H7E)+Chr$(&HA5)
      Su%(D%,16)=36
      For I%=1 To 128
        Mid$(D$(D%,24),I%)=
Chr$(&HFE)
      Next I%
      Su%(D%,24)=127
      Df%(D%)=2
      @Dinfo
    Endif
Return
Procedure Fmtsure
    D%=1+(Menu(0)<19)
    Sure%=1
    If Df%(D%)
      Alert 1," Formatieren
löscht alles ! !",1," OK
! Stop ",     Sure%
    Endif
    If Sure%=1
      @Init
    Endif
Return
Procedure Dinfo2
    For D%=0 To 1
      @Dinfo
    Next D%
Return
Procedure Dinfo
    If D%(D%)=0
      Infow D%*2+1," Inaktiv"
    Else
      If Df%(D%)=0
        Infow 2*D%+1," Aktiv :
unformatiert"
      Else
        If Df%(D%)=1
          Infow D%*2+1," Aktiv :
 Single density"
        Else
          Infow D%*2+1," Aktiv :
 Double Density"
        Endif
      Endif
    Endif
Return
```

Verwaltungsprogramm für das XL-ST-Interface (Schluß)

Für Bastler: Cursor-Tasten selbst gemacht

Bei den XL-Geräten sind von Atari vier Funktionstasten vorgesehen, die in den Computer nicht eingebaut wurden. Sie könnten jedoch im XL nützliche Funktionen übernehmen. Warum die Tasten also nicht selbst einbauen?

Dem aufmerksamen XL-Anwender ist sicher aufgefallen, daß beim Tastaturtest im Selbsttestprogramm des XL vier Tasten nicht angesprochen werden können. Diese vier befinden sich auf dem Bildschirm zusammen mit den Funktionstasten rechts über der Abbildung der normalen Tastatur. Sie sind mit <1>, <2>, <3> und <4> gekennzeichnet. Diese vier zusätzlichen Funktionstasten waren beim 1200XL, 1400XL und beim 1450XLD eingebaut, die in Deutschland nie zur Auslieferung gelangten. In den erhältlichen XL-Baureihen fehlen sie, obwohl das Betriebssystem sie unterstützt. Durch den Einbau dieser vier Tasten erhält man separate Cursor-Tasten. Im einzelnen haben diese vier Tasten folgende Funktionen:

<1> Cursor nach oben, entspricht <CONTROL> und <->
<2> Cursor nach unten, entspricht <CONTROL> und <=>
<3> Cursor nach links, entspricht <CONTROL> und <+>
<4> Cursor nach rechts, entspricht <CONTROL> und <*>

Zusammen mit <SHIFT>:
<1> Home (linke obere Ecke)
<2> linke untere Ecke
<3> Cursor an den linken Rand
<4> Cursor an den rechten Rand

Mit <CONTROL>:
<1> Tastatur abschalten, nochmaliges Drücken wieder anschalten.
<2> Bildschirm ausschalten (POKE 559,0). Beliebiger Tastendruck stellt wieder auf vorherigen Wert.
<4> Wechselt zwischen den beiden Zeichensätzen.

Warum sollte man sich diese nützlichen Funktionen entgehen lassen? Selbst einbauen heißt die Devise.

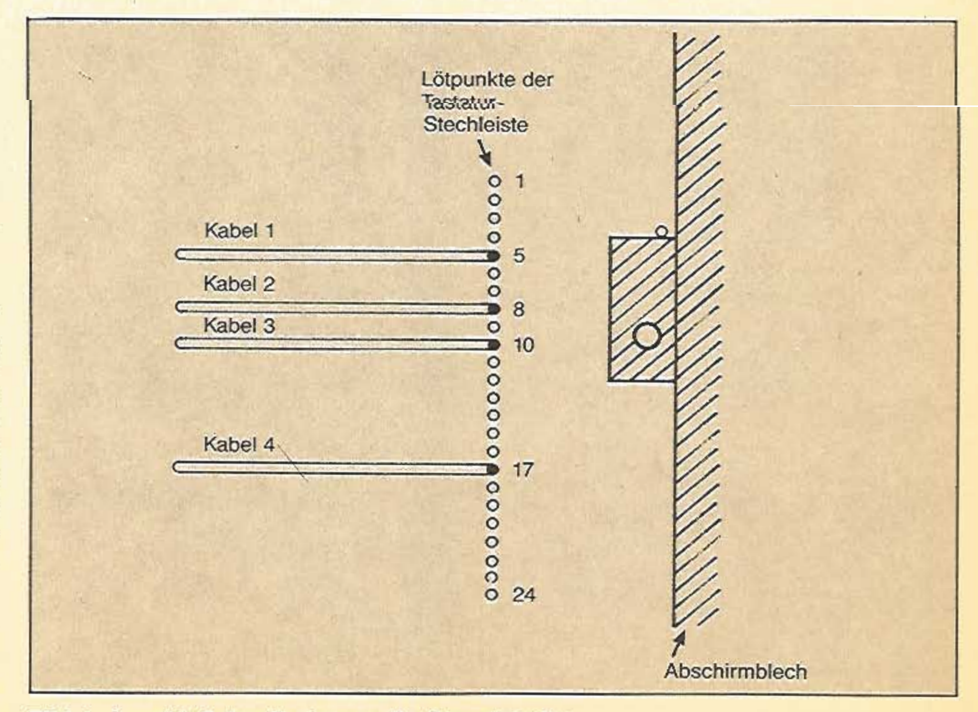


Bild 1. Anschluß der Tasten an die Grundplatine

Bevor Sie sich jedoch Lötkolben schwingend an ihrem Computer zu schaffen machen, sollten Sie bedenken, daß ein derartiger Eingriff den Verlust einer eventuell noch existierenden Garantie zur Folge hat. Solange man das Original-Betriebssystem des XL benutzt, hat man nur Vorteile. Andere Betriebssysteme, wie zum Beispiel das des alten 800ers (das mit dem Atari-Translator geladen wird), fragen diese Tasten nicht ab, womit diese ohne Funktion bleiben.

Doch nun frisch ans Werk, Kreuzschlitz-Schraubenzieher, Handbohrer oder ähnliches, Lötkolben, Lötzinn, vier Tasten und ein paar Zentimeter Litze bereitlegen. Wegen ihrer niedrigen Bauhöhe empfehlen sich Digi-Taster. Die Kosten hierfür betragen ungefähr vier Mark.

Um ein Zerkratzen des XL zu verhindern, ist es ratsam, sich eine weiche Unterlage zu beschaffen. Legen Sie nun den Computer mit der Unterseite nach oben auf die Unterlage, lösen Sie die sechs Schrauben und verwahren Sie diese gut. Dann drehen Sie den Computer vorsichtig um, so daß die Tastatur zu Ihnen zeigt, und klappen das Oberteil nach rechts. Achten Sie aber darauf, daß Sie die Folie mit den aufgedruckten Leiterbahnen zwischen Rechnerplatine und Tastatur nicht beschädigen. Nun ziehen Sie gleichmäßig und ganz ganz vorsichtig die Folie aus der Verbindungsleiste auf der Computerplatine. Jetzt entscheidet sich, ob der Einbau einfach oder umfangreich wird. Sehen Sie bei ihrem Computer unter der Tastatur eine Metallplatte, ist die Aufrüstaktion mit etwas mehr Arbeit verbunden (Bild 1). Ist die Tastaturplatine zu sehen, gestaltet sich der Einbau einfacher (Bild 2).

Neben dem Schacht ist Platz

Zuerst werden die Tasten in den Computer eingebaut. Es empfiehlt sich, sie auf der weißen Oberseite links oder rechts neben dem Modulschacht unterzubringen. Bohren Sie dann die Löcher für die Kontakte der Tasten und kleben Sie die Tasten auf. Dann verlöten Sie sie nach Bild 3. Achten Sie darauf, den mit eins bis vier markierten Litzen eine ausreichende Länge zu geben, da diese mit der Tastatur beziehungsweise mit der Computerplatine verlötet werden. Diejenigen, bei denen unter die Tastatur eine Metallplatte geschraubt ist, haben noch weitere Arbeit vor sich, die anderen können den folgenden Absatz überspringen.

Zuerst müssen Sie zwei weitere Schrauben lösen. Die eine sitzt schräg links vor der Buchse, an die die Peripheriegeräte wie zum Beispiel Diskettenstationen des XL-Computers angeschlossen werden. Die andere befindet sich schräg rechts vor dem Stecker der Stromversorgung. Greifen Sie nun unter die Platine, drücken Sie auf den Einschaltknopf und schieben Sie so die Platine leicht aus der Verankerung. Durch eine Drehung nach rechts unten, wobei Sie einen leichten Druck auf die Diskettenbuchse ausüben, springt die Platine heraus. Jetzt drehen Sie die Platine um und lösen alle noch verbliebenen Schrauben des Abschirmblechs. Wo Sie nun die vier Kabel anlöten müssen, sehen Sie in Bild 1. Es ist ratsam die Kabel nicht direkt anzulöten, sondern einen Zwischenstecker zu verwenden, da die Tastatur sonst nicht mehr von der Computerplatine getrennt werden kann. Wenn Sie dünne Litze verwenden, können Sie die Abschirmung wieder zusammenschrauben. Verwenden Sie die obere Abdeckung nicht, müssen Sie den kleinen Draht im Oberteil des Gehäuses von der Klappe des Modulschachts entweder abkneifen oder so verbiegen, daß er keinen Kontakt zur Platine hat. Dann setzen Sie die Platine wieder in das Unterteil ein und verschrauben sie vorsichtig.

Wenn man die Kabel an der Tastaturplatine verlötet hat, unterführt man sie einfach wie in Bild 2 unter der Platine und lötet sie dort an. Ein Zwischenstecker entfällt hier, da ein Losschrauben der Tastatur von der Gehäuseoberseite wohl in den seltensten Fällen nötig wird.

Ist alles fertig, brauchen Sie nur noch das Tastaturkabel wieder einzusetzen und das Ober- und Unterteil des Gehäuses wieder zu verschrauben.

Nun müssen Sie endlich nicht mehr umständlich die <CONTROL>-Tasten drücken, um den Cursor über den Schirm zu steuern. Wenn Sie sich erst einmal an diese Vereinfachung gewöhnt haben, möchten Sie sie nicht mehr missen.

(Kjersten Waldheim/rj/hf)

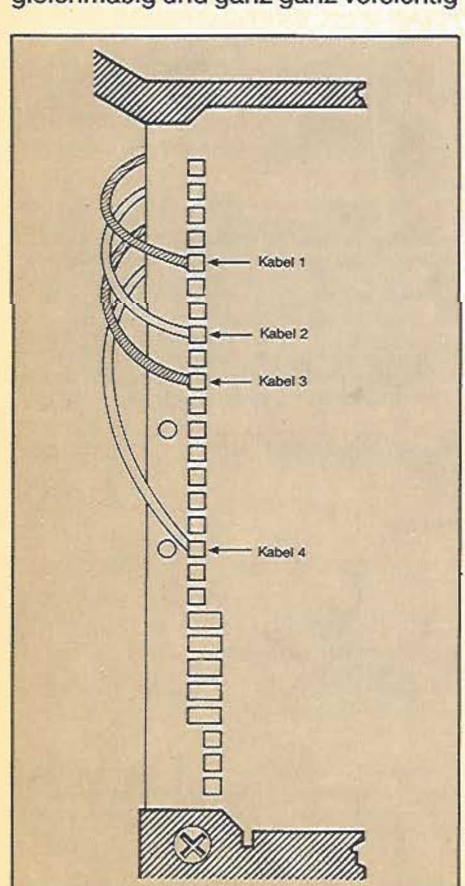


Bild 2. Unterseite der Tastatur

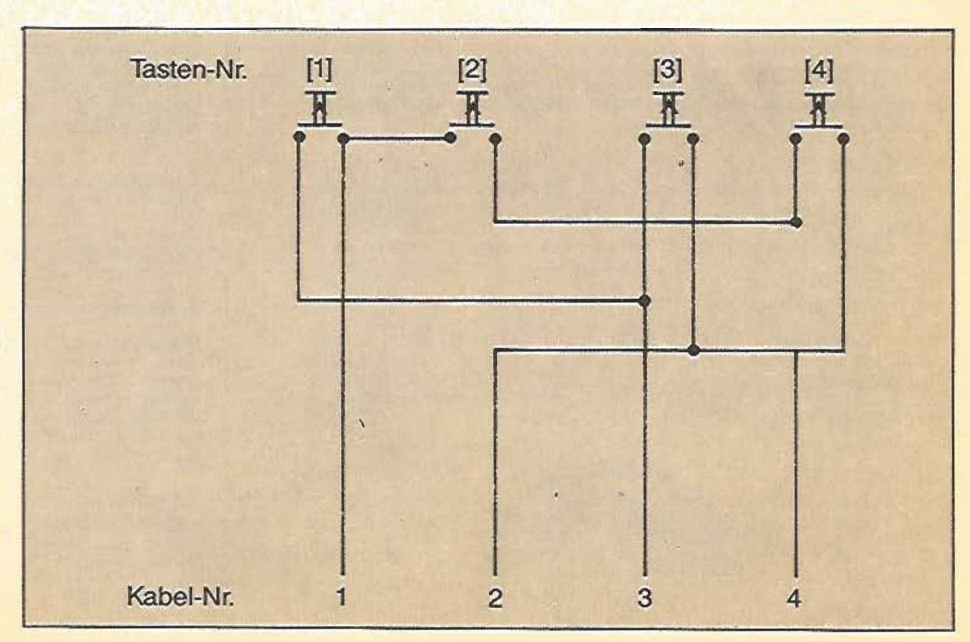


Bild 3. Verdrahtet werden die zusätzlichen Tasten nach obenstehendem Schema

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Michael Lang (lg)
Redakteure: Henrik Fisch (hf, Projektleitung), Joachim Graf (jg), Thomas Kaltenbach (kl), Udo Reetz (rz), Richard Joerges (rj), Thomas Jacobi (ja), Anatol Locker (al), Gregor Neumann (gn);
Chef v. Dienst: Petra Wängler
Redaktionsassistenz: Monika Lewandowski (222), Rita Gietl (289)

Fotografie: Jens Jancke

Titelgestaltung: Katja Milles

Layout: Leo Eder (Ltg.),
Katja Milles, Andrea Miller

Produktionsleiter: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph-Peter Rauchfuss

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG,
Kollerstrasse 3, CH-6300 Zug,
Tel. (042) 415656, Telex: 862329 mut ch
USA: M&T Publishing Inc., 501 Galveston Dr., Redwood City, CA 94063;
Tel. 415-366-3600, Telex 752-351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Anzeigenverkauf: Alicia Clees (313), Britta Fiebig (211)

Anzeigenverwaltung und Disposition:
Patricia Schiede (172)

Marketingleiter: Hans Hörl (114)

Vertriebsleiter: Helmut Grünfeldt (189)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Str. 96, 7000 Stuttgart 1, Tel. (0711) 6483-0

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Bezugspreis: Das Einzelheft kostet DM 14,–

Druck: SOV St. Otto-Verlag GmbH,
Laubanger 23, 8600 Bamberg

Verantwortlich:
Für redaktionellen Teil: Michael Lang
Für Anzeigen: Alicia Clees

Redaktionsdirektor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, 8013 Haar bei München,
Telefon (089) 4613-0, Telex 5-22052

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen (089) 4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

ISSN 0931-5829



NEU!

von





Daß die 8-Bit-Ataris vierstimmigen Sound erzeugen können, ist ja bekannt. Aber was halten Sie von echtem Synthesizerhall, freier Ton- und Hüllkurvengestaltung, Schlagzeugsound und Transponierautomatik?

Wenn Sie jetzt glänzende Augen bekommen, ist MASIC das Richtige für Sie. Eine strukturierte Programmiersprache speziell für Musik und Sound. Und, besonders wichtig: Das, was MASIC hervorbringt, läßt sich mühelos für eigene Programme nutzen. MASIC-Programme arbeiten im Interrupt, ohne z.B. ein gleichzeitig laufendes Basic-Programm zu stören.

Zeigen Sie anderen Heimcomputer-Musikfreunden, was eine Harke ist – mit MASIC!

Disk DM 49,-

Da haben Sie gerade ein herrliches Bild entdeckt, wie geschaffen als Titelgrafik für Ihr Basic-Programm. Oder Sie würden zu gern einmal mit einem Malprogramm ein wenig am Portrait dieser Dame auf der Illustrierten-Titelseite herummanipulieren.

Sie haben einen 8-Bit-Atari und einen Epson-FX-80- kompatiblen Drucker? Dann halten Sie sich fest, denn scantronic zaubert ein Abbild Ihrer Papiervorlagen mit Hilfe Ihres Druckers in Graphics 9 auf den Bildschirm und/oder die Diskette.

scantronic, unser Scanner, wird anschlußfertig geliefert, komplett mit Steuersoftware (Disk), einem fähigen Graphics-9-Malprogramm in Turbo-Basic XL, 16-Graustufen-Hardcopy und deutscher Anleitung.



DM 59,-

Endlich! Zwei neue deutsche Textadventures, die Sie begeistern werden!

Alptraum

Sie schlüpfen in die Rolle des Besitzers einer kleinen Fluglinie. Sein nächtlicher Alptraum, in dem sich nach und nach die bedrohlichen Gefahren herauskristallisieren, ist Ihr Spiel. Je mehr Sie im Traum erfahren, desto besser. Je schneller der Traum zu Ende ist, desto quälender wird die zurückbleibende Ungewißheit sein.

Der leise Tod

Sie sind Ray Cooper, von Beruf Privatdetektiv, und besitzen ein kleines Büro in der Themsestadt London. Eines Tages erhalten Sie ein Telegramm von einem guten Freund im fernen Amerika. Ein heikler und gefährlicher Auftrag wartet auf Sie. Setzen Sie sich in Ihren Ferrari und brechen Sie auf – ob Sie jemals nach London zurückkehren, können wir Ihnen nicht sagen.

Beide Adventures laufen auf allen Computern der Atari-XL- und XE-Serie ab 64 KByte RAM.
Pro Diskette **DM 39,-**



Software-Bestellschein

Ich bestelle aus dem Diabolo-Versand folgende Software:

HCSA

Anzahl	Titel	Gesamtpreis

Computertyp

Name des Bestellers

Anschrift

PLZ/Ort

Datum/Unterschrift

Ich wünsche folgende Bezahlung:
- ☐ Nachnahme (zuzüglich 5.70 DM Versandkosten)
- ☐ Vorauskasse (zuzüglich 3.– DM Versandkosten, ab 100 DM Bestellwert versandkostenfrei)

Bei Vorauskasse bitte Scheck beilegen.
Software ist vom Umtausch ausgeschlossen.

Coupon ausschneiden, auf Postkarte kleben und einsenden an:
R + E Software, PF 1640, 7518 Bretten.
Eine Abteilung des Verlags Rätz-Eberle GdbR.

Der neue ATARI 800 XE ist soeben gelandet

Der neue ATARI 800 XE mit Supergrafik ist jetzt da. Der Computer für alle, die erster Klasse fliegen wollen.

Das sind die technischen Details des neuen Spitzenmodells der 64er Klasse:
INNENRAUM: 64 kBytes Arbeitsspeicher im RAM
COCKPIT: 24 kBytes. ROM-resident.
Für Betriebssystem und Basic.
ZENTRALTRIEBWERK: 6502 Prozessor der C-Version

ZUSATZTRIEBWERKE:
POKEY – für den Sound
GTIA – für die Grafik
ANTIC – für den Bildschirm
PIA – für die Ein-/Ausgabe
SCHUB: 1,79 Megahertz
ZULADUNG: Datasette, Floppy, Drucker, Monitor, Fernseher, Joystick, Lichtgriffel, Maltablett, Modulschacht.
SPANNWEITE: Musik, Grafik, Spiele, Textverarbeitung, Kalkulation, Lernhilfe, Archivierung, Verwaltung und, und, und ...

Das sind viele tausend weitere Programme der erfolgreichen ATARI XE/XL-Serie.
ATARI 800 XE mit Supergrafik. Das ist Computertechnologie für Menschen von heute zum sagenhaft niedrigen Preis.
Den neuen ATARI 800 XE gibt es ab jetzt im Handel.



... wir machen Spitzentechnologie preiswert.